The microtype package

Subliminal refinements towards typographical perfection

R Schlicht
w.m.l@gmx.net

v2.8
2020/12/07

The `microtype` package provides a LaTeX interface to the micro-typographic extensions that were introduced by pdfTeX and have since also propagated to LuaTeX and XeTeX: most prominently, character protrusion and font expansion, furthermore the adjustment of interword spacing and additional kerning, as well as hyphenatable letterspacing (tracking) and the possibility to disable all or selected ligatures. These features may be applied to customisable sets of fonts, and all micro-typographic aspects of the fonts can be configured in a straight-forward and flexible way. Settings for various fonts are provided.

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

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

Contents

List of Tables

1 Micro-typography with TEX

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

Some essential micro-typographical aspects are already taken care of by TEX out of the box – and in an outstanding manner – namely, hyphenation and justification, as well as kerning and ligatures. Other aspects are in the user's scope of responsibilities, e.g., to specify the right amounts of spacing around punctuation characters, numbers, or quotation marks. On top of this, a number of long-standing micro-typographic techniques have been introduced to the TEX world relatively recently with pdfTEX, and have since also propagated to LuaTEX and XƎTEX. These features make them the tool of choice not only for the creation of electronic documents but also of works of outstanding time-honoured typography: most prominently, *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Thế Thành, the author of pdfTEX, who writes in his thesis:

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

Protrusion	*off*
Expansion	*off*

Both features are enabled throughout this document.

> 'Margin kerning is the adjustments of the characters at the margins of a typeset text. A simplified employment of margin kerning is hanging punctuation. Margin kerning is needed for optical alignment of the margins of a typeset text, because mechanical justification of the margins makes them look rather ragged. Some characters can make a line appear shorter to the human eye than others. Shifting such characters by an appropriate amount into the margins would greatly improve the appearance of a typeset text.
>
> Composing with font expansion is the method to use a wider or narrower variant of a font to make interword spacing more even. A font in a loose line can be substituted by a wider variant so the interword spaces are stretched by a smaller amount. Similarly, a font in a tight line can be replaced by a narrower variant to reduce the amount that the interword spaces are shrunk by. There is certainly a potential danger of font distortion when using such manipulations, thus they must be used with extreme care. The potentiality to adjust a line width by font expansion can be taken into consideration while a paragraph is being broken into lines, in order to choose better breakpoints.' [Thành 2000, p. 323]

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

Setting *additional kerning* for individual characters is especially (but not only) useful for languages whose typographical tradition requires certain characters to be separated by a space. For example, it is customary in French typography to add a small space before question mark, exclamation mark and semi-colon, and a bigger space before the colon and the guillemets. Until now, this could only be achieved

---

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

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

Adjustment of interword spacing is based upon the idea that in order to achieve a uniform greyness of the text, the space between words should also depend on the surrounding characters. For example, if a word ends with an 'r', the following space should be a tiny bit smaller than that following, say, an 'm'. You can think of this concept as an extension to TEX's 'space factors'. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word 'may': this extension is still highly experimental – in particular, only ending characters will currently influence the interword space. Also, the settings shipped with `microtype` are but a first approximation, and I would highly welcome corrections and improvements. I suggest reading the reasoning behind the settings in section 15.9.

The possibility, finally, to *disable all or selected ligatures* is particularly useful for typewriter fonts.

The `microtype` package provides an interface to all these micro-typographic extensions. All micro-typographic aspects may be customised to your taste and needs in a straight-forward and systematic manner. The next chapters present a survey of all options and customisation possibilities. Should the micro-typographic extension discussed in a section work only with certain TEX engines, this requirement is marked inside a grey text box on the right.

2 Getting started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

This will be sufficient in most cases, and if you are not interested in fine-tuning the micro-typographic appearance of your document (however unlikely this would seem, since using this package is proof of your interest in typographic issues), you may actually skip the rest of this document. If this, on the other hand, does not satisfy you – be it for theoretical or practical reasons – this manual will guide you on the path to the desired results along the following milestones:

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

3 Options

Like many other LaTeX packages, the `microtype` package accepts options in the well-known `key=value` syntax. In the following, you will find a description of all `keys` and their possible `values` (‘`true`’ may be omitted; multiple values, where allowed, must be enclosed in braces; the default value is shown on the right – if preceded by an asterisk, this default only applies when running an up-to-date pdfTeX in PDF mode).

3.1 Enabling the micro-typographic features

`protrusion` `expansion` `true`, `false`, `compatibility`, `nocompatibility`, ⟨*font set name*⟩ * `true`

These are the main options to control the level of micro-typographic refinement which the fonts in your document should gain. By default, the package is moderately greedy: character protrusion will always be enabled, font expansion will only be disabled when the fonts cannot be expanded automatically, that is, with pdfTeX versions older than 1.20 or in DVI output mode (see section 3.5), or with XeTeX. In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (hence, it is usually not necessary to load the package with different options, e.g., for PDF resp. DVI mode).

`activate`

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

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

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

```
\usepackage{microtype}
```

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

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

`tracking` `true`, `false`, ⟨*font set name*⟩ `false`

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

`kerning` `spacing` `true`, `false`, ⟨*font set name*⟩ `false`

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

Table 1: Availability of micro-typographic features

TeX engine			Micro-typographic features					
Engine	Version	Output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
pdfTeX	< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
	≥ 0.14f	DVI/PDF	★	⊠	∅	∅	∅	∅
	≥ 1.20	DVI	★	⊠	∅	∅	∅	∅
		PDF	★	⊠	★	∅	∅	∅
	≥ 1.40	DVI	★	⊠	∅	⊠	⊠	∅
		PDF	★	⊠	★	⊠	⊠	⊠
LuaTeX	≥ 0.30	DVI	★	⊠	∅	∅	∅	∅
		PDF	★	⊠	★	∅	∅	∅
	≥ 0.62	DVI	★	∅	(⊠)[a]	∅	∅	⊠
		PDF	★	∅	★	∅	∅	⊠
XƎTeX	≥ 0.9997	PDF	★	∅	∅	∅	∅	∅

★ = enabled ⊠ = not enabled ∅ = not available [a] by means of variable tracking

Table 1 presents an overview of which micro-typographic features are available and enabled by default for the relevant TeX versions and output modes.

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

3.2 Character protrusion

pdfTeX 0.14f | LuaTeX 0.30 | XƎTeX 0.9997

`factor` ⟨*integer*⟩ 1000

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

`unit` `character`, ⟨*dimension*⟩ `character`

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

3.3 Font expansion

pdfTeX 0.14f | LuaTeX 0.30

`auto` `true`, `false` * `true`

Beginning with pdfTeX version 1.20 (inherited by LuaTeX), the expanded instances of the fonts may be calculated automatically and at run-time instead of the user having to prepare them in advance. This option is true by default provided that you are using a TEX engine with this capability and the output mode is PDF.[2] If `auto` is set to false, the font instances for all expansion steps must exist (with files called ⟨*font name*⟩±⟨*expansion value*⟩, e.g., `cmr12+10`, as described in the pdfTEX manual). With LuaTEX, expansion is always automatic.

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

`stretch` ⟨*integer*⟩ 20
`shrink`

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

`step` ⟨*integer*⟩ * 1

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

`selected` `true`, `false` `false`

When applying font expansion, it is possible to restrict the expansion of some characters that are more sensitive to deformation than others (e.g., the ‘O’, in contrast to the ‘I’). This is called *selected expansion*, and its usage allows increasing the stretch and shrink limits (to, say, 30 instead of 20); however, the gain is limited since at the same time the average stretch variance will be decreased. Therefore, this option is by default set to `false`, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.

3.4 Tracking

pdfTEX 1.40 | LuaTEX 0.62

`letterspace` ⟨*integer*⟩ 100

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

---

[2] With pdfTEX, automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in `T1` encoding, you should either install the `cm-super` package or use the Latin Modern fonts (package `lmodern`).

[3] The downside with this default is that pdfTEX may run out of memory with huge documents; in this case, read about the error messages in the ‘Hints and caveats’ section (9), or try with a larger `step`.

3.5 Miscellaneous options

`draft` `true`, `false` `false`

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

`verbose` `true`, `false`, `errors`, `silent` `false`

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

`babel` `true`, `false` `false`

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

`config` ⟨*file name*⟩ `microtype`

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

`DVIoutput` `true`, `false` * `false`

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

3.6 Changing options later

`\microtypesetup` { ⟨*key* = *value list*⟩ }

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

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

4 Selecting fonts for micro-typography

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

`\DeclareMicrotypeSet` [⟨*features*⟩] {⟨*set name*⟩} {⟨*set of fonts*⟩}

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

The set of fonts is specified by assigning values to the NFSS font attributes: encoding, family, series, shape and size (cf. LaTeX 2$_{\varepsilon}$ font selection). Let's start with an example. In the main configuration file `microtype.cfg`, a font set called '`basictext`' is defined as follows:

```
\DeclareMicrotypeSet{basictext}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
     family   = {rm*,sf*},
     series   = {md*},
     size     = {normalsize,footnotesize,small,large}
   }
```

If you now call

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

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

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

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

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

If a value is followed by an asterisk (like '`rm*`' and '`sf*`' in the first example), it does not designate an NFSS code, but will be translated into the document's `\`⟨*value*⟩`default`, e.g., `\rmdefault`.[4] A single asterisk means `\`⟨*attribute*⟩`default`, e.g., `\encodingdefault`, respectively `\normalsize` for the size axis. Sizes may either be specified as a dimension ('`10`' or '`10pt`'), or as a size selection command *without* the backslash. You may also specify ranges (e.g., '`small-Large`'); while the lower

---

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

Table 2:

Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
all	∅	∅	∅	∅	∅
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	∅	∅	∅	∅
alltext-nott (allmath-nott)	Text encodings, TS1 (OML, OMS, U)	\rm*, \sf*	∅	∅	∅
basictext (basicmath)	Text encodings (OML, OMS)	\rm*, \sf*	\md*	∅	\normalsize, \footnotesize, \small, \large
smallcaps	Text encodings	∅	∅	\sc*,si,scit	∅
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, EU1, EU2, TU '\...*' = '\...default'

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

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

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

As you can tell from the example, the asterisk notation is also permitted for the `font` key. A single asterisk is equivalent to '*/*/*/*/*', i.e., the normal font. Size selection commands are possible, too, however, ranges are not allowed.

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

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

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

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

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

If a feature is enabled but no font set has been chosen explicitly, the sets declared by this command will be activated. By default, the '`alltext`' font set will be activated for character protrusion and additional kerning, the '`alltext-nott`' set for font expansion and interword spacing, and the '`smallcaps`' set for tracking.

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

5 Micro fine tuning

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

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

The set of fonts to which the settings should apply is declared using the same syntax of ⟨*font axis*⟩ = ⟨*value list*⟩ pairs as for the command `\DeclareMicrotypeSet` (see section 4), with the only difference that values including asterisks (which, as you may recall, stand for the respective default) will be translated immediately instead of at the end of the preamble. To find the matching settings for a given font the package will try all combinations of font encoding, family, series, shape and size, with decreasing significance in this order. For instance, if settings exist for both the current family (say, `T1/cmr///`) and for italic fonts in the normal weight (`T1//m/it/`), the settings for the `cmr` family would apply. The encoding must always match.

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

5.1 Character protrusion

pdfTEX 0.14f | LuaTEX 0.30 | XƎTEX 0.9997

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

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

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

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

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

Options:

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

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

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

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

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

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

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

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

`context` The scope of the list may be limited to a certain context. For further details, see section 6.

5.2 Font expansion

pdfTEX 0.14f | LuaTEX 0.30

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

By default, all characters of a font are allowed to be stretched or shrunk by the same amount. However, it is also possible to limit the expansion of certain characters if they are more sensitive to deformation. This is the purpose of the `\SetExpansion` command. Note that it will only have an effect if the package has been loaded with the `selected` option (cf. section 3.3). Otherwise, the expansion settings will be ignored – unlike the options in the optional first argument, which will still be evaluated. If the `selected` option has been set to true, and settings for a font don't exist, font expansion will not be applied to this font at all. Should the extraordinary situation arise that you want to employ selected expansion in general but for a particular font (set) all characters should be expanded or shrunk by the same amount, you would have to declare an empty list for these fonts.

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

Options:

`name`, `load`, `preset`, `inputenc`, `context` Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset

---

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

all expansion factors, to set the input encoding, or to determine the context of the list (expansion contexts are only possible with pdfTeX version 1.40.4 or newer).

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

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

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

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

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

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

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

5.3 Tracking

pdfTeX 1.40 | LuaTeX 0.62

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

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

---

[6] For older versions, a dirty trick is laid out in section 14.2 on page 58.

to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.[7] The `\SetTracking` command allows specifying the tracking amount for different fonts or font sets. It will also be evaluated by the `\textls` command, which may be used for letterspacing shorter pieces of text (see section 7).

The tracking amount is specified in thousandths of 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 letterspaced 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 default equal to the current letterspace amount. To remove kerning on both sides, you would write '`outer kerning={0,0}`'.

`no ligatures` By default, ligatures in letterspaced fonts will be constructed as usual, which may be advisable when changing the tracking by only a small amount. For larger letterspacing amounts, on the other hand, the normal letter space within ligatures would have displeasing effects. This key expects a comma-separated list of characters for which ligatures should be disabled; only the character that begins a ligature must be specified. If the key is given without a value, *all* ligatures of the font will be disabled. With pdfTEX, this is not recommended, however, since it entails that kerning will be switched off, too. With LuaTEX, there is no such limitation. The default settings disable ligatures for the character 'f' only, i.e., 'ff',

---

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

‘fi’, ffi’, etc.[8] In exceptional situations, you can manually break up a ligature by inserting ‘`{\kern0pt}`’ resp. `babel`’s `"|` shortcut, or protect it by enclosing it in `\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 are in no way recommended; they only serve illustrative purposes):

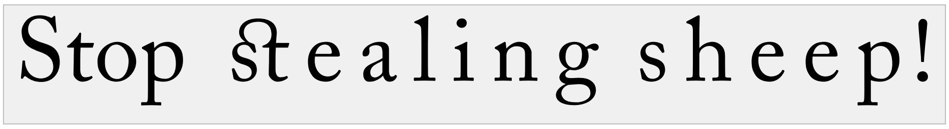
```
\SetTracking
  [ no ligatures  = {f},
    spacing       = {600*,-100*, },
    outer spacing = {450,250,150},
    outer kerning = {*,*} ]
  { encoding = * }
  { 160 }
```

and then write:

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

this would be the (typographically dubious) outcome:

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.



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 em = 0.16 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.

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. This could be achieved with the following settings:

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

Letterspaced fonts for which settings don’t exist will be spaced out by the default of 0.1 em (adjustable with the package option `letterspace`, see section 3.5). Suppose

---

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

your editor wants you to shorten your 1000-pages chef-d'œuvre by a handful of pages, you could load `microtype` with (fingers crossed):

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

5.4 Additional kerning

pdfTEX 1.40

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

With this command, you can fine tune the extra kerning. In contrast to standard kerning, which is always associated with a *pair* of characters, and to tracking, which specifies the space between *all* characters of a font, the extra kerning relates to single characters, that is, whenever a particular character appears in the text, the specified kerning will be inserted, regardless of which character precedes resp. follows it. (Put differently, this feature allows modifying the left or right *sidebearings* of specific glyphs.)

It should not be neglected to mention a limitation of this feature: words *immediately following* such a kern (not separated by a space) will not be hyphenated, unless you insert the breakpoints manually, e.g., for kerning after the apostrophe, '`l'apos\-trophe`'. Furthermore, additional kerning will not be applied in math mode. These restrictions of pdfTEX will hopefully be lifted some time.

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

Options:

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

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

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

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

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

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

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

5.5 Interword spacing

pdfTEX 1.40

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

This command allows you to fine tune the interword spacing (also known as glue). A preliminary remark on what a 'space' is may be in order: between two words, 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. pdfTEX's additional spacing adjustment may be considered as an extension to space factors with much finer control: while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – you may modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. Note that when interword spacing adjustment is in effect, space factors are ignored.

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

Options:

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

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

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

the space inserted after a full stop would be doubled (technically speaking: 2 × `\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 Ă should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list of features, you can confine the scope of the list. Additionally, it accepts the `inputenc` key to set the input encoding for this list. The font set can be declared in the usual way. The inheritance lists are declared as pairs of ⟨*base character*⟩ = ⟨*list of inheriting characters*⟩. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

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

5.7 Configuration files

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

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

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

`\DeclareMicrotypeVariants*`

On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a 'variant' of the base font (cf. Karl Berry's Fontname). It is thus possible to put settings for, e.g., the fonts `padx` (expert set), `padj` (oldstyle numerals) and `pad` (plain) into one and the

Table 3:

Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
	Encodings [Scripts]	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, T2A, 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
Latin Modern Roman[i]	EU1/2, TU [Latin, Greek]	n, it, (sl)[d]
Charis SIL	EU1/2, TU [Latin, Cyrillic, Greek]	n, it, sc
Palatino Linotype[j]	EU1/2, TU [Latin]	n, it, sc
Computer Modern math (`cmsy`, `cmm`)[k]	OML/OMS	n/it
AMS symbols (`msa`, `msb`)	U	n
Euler (`eur`, `eus`, `euf`)[l]	U	n
Euro symbols (Adobe, ITC, `marvosym`)	U/OT1	n, it

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

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.[9] This command will load the file '`mt-`⟨*font name*⟩`.cfg`'.

6 Context-sensitive setup

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

`\microtypecontext` {⟨*context assignments*⟩}

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

`\begin{microtypecontext}` {⟨*context assignments*⟩}
`\end{microtypecontext}` Like many LaTeX commands, it is also available in the form of an environment.

`\textmicrotypecontext` {⟨*context assignments*⟩} {⟨*general text*⟩}

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

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

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

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

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

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

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

---

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

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` {⟨*list of* `babel` *languages*⟩} {⟨*context list*⟩}

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

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

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

7 Letterspacing revisited

pdfTEX 1.40 | LuaTEX 0.62

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

While the `tracking` feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.[10] For such ad-hoc letterspacing, `microtype` introduces two commands that can be used (independently of whether the `tracking` option is enabled) in the same way as LATEX's text commands: `\textls` – which also works
`\lsstyle` 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.
`\textls*` The starred version of `\textls` does not add any extra kerning before or after the text, which may be useful, e.g., for section titles. By default, each character will be spaced out by 100/1000 em = 0.1 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.

---

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

`\lslig` {⟨*ligature*⟩}

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

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

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{120}
\textfrak{\lsstyle Au\lslig{s:}si\lslig{ch}t\lslig{s:}losigkeit}
```

`letterspace.sty` These three commands (plus the `letterspace` option, described in section 3.4) are also available with the alternative `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions (and also omitting the possibilities of the `\SetTracking` command – all ‘f’ ligatures will be disabled, inner and outer spacing and outer kerning will be set to the default values described in section 5.3). If you prefer to forgo `microtype`’s specialties, you may load the `letterspace` package instead. Both packages should not be used at the same time.

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

8 Disabling ligatures

pdfTeX 1.30 | LuaTeX 0.30

`\DisableLigatures` [⟨*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.
```

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

9 Hints and caveats

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

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

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

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

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

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`

---

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

environment. However, `microtype` has no knowledge about the context that a font appears in but will solely decide by examining its attributes. Therefore, you have to take care of disabling protrusion in `verbatim` environments for yourself (that is, if you don't want to disable protrusion for the typewriter font altogether, by activating, say, the font set '`alltext-nott`'). While the `\microtypesetup` command has of course been designed for cases like this, you may find it tiresome to repeat it every time if you are using the `verbatim` environment frequently. The following line (which requires the `etoolbox` package), added to the document's preamble, would serve the same purpose:

```
\AtBeginEnvironment{verbatim}{\microtypesetup{activate=false}}
```

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

Settings for Greek/Thai/Armenian etc. encodings are not yet included. The default sets of fonts for which the micro-typographic features will be enabled (see table 2) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e.g., LGR, T2B, etc.), `microtype` will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion at least, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

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

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

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

- Even though all configuration files are still provided in legacy (7-bit) format, using multi-byte (Unicode) characters in the settings should run smoothly with an up-to-date LATEX system. For older systems or documents in legacy encodings, in contrast, this requires loading the `inputenc` package first. Furthermore, when using multiple input encodings in a document, 8-bit characters in the settings will only work reliably if you specify the `inputenc` key.

- When loading the package with the `babel` option, you must load the `babel` package before `microtype`.
- Before this package was fully compatible with LuaTeX, the following method of enabling expansion and protrusion with the `fontspec` package was most often found to be recommended:

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

  This code should *not* be used with this package, as it will basically override all of the settings made by `microtype` – despite the naming, the above lines have nothing to do with this package.[12]
- With pdfTeX, it is currently not possible to create character-specific settings for Chinese/Japanese/Korean fonts. Therefore, the only micro-typographic extension that can be made to work with `CJK` fonts is (non-selected) font expansion.
- When used with the `xeCJK` package or the `luatexja` package, text commands (e.g., `\'A`, `\textless`) in the configuration will not be understood. You therefore have to ensure that `microtype` will encounter none of them. This requires, firstly, that the glyphs be specified only as single (possibly Unicode) characters, as numbers, or as glyph names (cf. section 5); and secondly, if you are using a font for which pre-defined settings do not exist, that you create these settings yourself (because otherwise, the default settings will be loaded, which do contain text commands). Furthermore, you should load `microtype` late.

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

- `! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.`

  This error message will occur if you are trying to employ font expansion while creating DVI output. Remember that *automatic* font expansion only works when running pdfTeX or LuaTeX in PDF mode. Although expansion is also possible in DVI mode with pdfTeX, it requires that all instances of the expanded fonts exist on your TeX system.
- `! pdfTeX error (font expansion): auto expansion is only possible with scalable fonts.`

  Automatic font expansion has been improved in pdfTeX 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (`pk`) font, which is still not possible, or that the font isn't found at all, e.g., because of missing map entries.
- `Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)`

  and the PDF viewer complains about a missing font, e.g., Adobe Reader thusly:

  `Could not find a font in the Resources dictionary - using Helvetica instead.`

  With pdfTeX versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your TeX system is not set up to embed (or 'download') the base PostScript fonts (e.g., Times, Helvetica, Courier). In most TeX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to `true`.

---

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

- `Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found`

  Furthermore, pdfTEX 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 pdfTEX 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.

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

10 Contributions

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

11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn't created the 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, Thành 2008 and elsewhere. I also thank him and the rest of the pdfTEX team, and more recently also the LuaTEX and XƎTEX teams, for refuting the idea that TEX is dead, and for fixing the bugs I find.

Harald Harders has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. *Andreas 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. *Georg Duffner* has patiently tested `microtype` under XƎTEX and LuaTEX with his beautiful OpenType font EB Garamond[13]. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding, as well as to *Karl Karlsson* for providing settings for the Cyrillic T2A encoding, and to *Hendrik Vogt*, who made substantial improvements to the Computer Modern Roman italic settings. I thank *Loren B. Davis* for providing protrusion settings for OpenType versions of Palatino Linotype. I am also very much indebted to *Élie Roux*, who not only contributed the `lua` module in the first place, but also, together with *Philipp Gesang*, took care of updating it for the developments in LuaTEX land.

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

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order, quotes indicate TeX.SX user names): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulle*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dyballa*, *Morten Høgholm*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis*, *Sven Naumann*, *Ross Hetherington*, *Wiebke Petersen*, *Geoff Vallis*, *Steven E. Harris*, *Karl Berry*, *Peter Meier*, *Nathan Rosenblum*, *Wolfram Schaalo*, *Vasile Gaburici*, *Sveinung Heggen*, *Axel Berger*, *Colin Rourke*, *Maverick Woo*, *Silas S. Brown*, *Lars Rönnbäck*, *Christian Stark*, *Leo*, *Marcin Borkowski*, *hscm*, *George Gratzer*, *Josep Maria Font*, *Juan Acevedo*, *Heiko Oberdiek*, *Till A. Heilmann*, *Rolf Dieterich*, *Seamus Bradley*, *Meho R*, *Steffen Hoffmann*, *Scott Pakin*, *Maïeul Rouquette*, *Jonas Hogstrom*, *Gabriel Kerneis*, *'RazorXsr'*, *Sebastian Schubert*, *'Dave'*, *Giuseppe Palma*, *Stephan Stiller*, *Christopher Schramm*, *'uli'*, *Sam Mason*, *'kleenstar'*, *'Henning'*, *Ronnie Marksch*, *David Carlisle*, *'Max'*, *'HcN'*, *Will Robertson*, *'user11126'*, *Ulrike Fischer*, *'Daniel'*, *'ltcomdata'*, *Reinhard Kotucha*, *'jcr'*, *Nils Anders Danielsson*, *Paolo Ney*, *Frank Mittelbach*, *Franz Wexler*, *Moritz Wemheuer*, *'Andy N'*, *Phelype Oleinik*, *Falk Hanisch*, *Markus Kohm*, *Paolo Polesana*, *Oliver Kopp*, *Hironori Kitagawa*, *'d909'*, *Daniel Benjamin Miller*, *Md Ayquassar* and *Marcel Krüger*.

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`)

---

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

Hàn Thế Thành, 'Micro-typographic extensions of pdfTeX 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 pdfTeX', in: *TUGBoat*, vol. 29 (2008), no. 1: 'EuroBachoTeX 2007 Proceedings', pp. 36–41. (Online at `http://www.tug.org/TUGboat/Articles/tb29-1/tb91thanh-fonts.pdf`)

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

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

LaTeX3 Project Team, *LaTeX 2$_{\varepsilon}$ font selection*, July 2020. (Available from CTAN at `/macros/latex/base/fntguide.pdf`)

Will Robertson, *The `fontspec` package: Font selection for XeLaTeX and LuaLaTeX*, 21 February 2020. (Available from CTAN at `pkg/fontspec`)

Élie Roux, Khaled Hosny, Philipp Gesang, Ulrike Fischer, Marcel Krüger, *The `luaotfload` package*, 2 September 2020. (Available from CTAN at `pkg/luaotfload`)

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

Melchior Franz, *The `soul` package*, 17 November 2003. (Available from CTAN at `pkg/soul`). See also Heiko Oberdiek's extension of this package, `soulutf8`, which adds Unicode support. (Available from CTAN at `pkg/soulutf8`)

13 Short history

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

2.8 (2020/12/07)

- New default font sets for expansion and spacing: '`alltext-nott`' [4, table 2]

2.7 (2017/07/07)

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

2.6 (2016/05/01)

- Support for LuaTeX ≥ 0.85
- Improvements for tracking/letterspacing with LuaTeX (`Renderer=Basic` no longer required)
- New font sets: '`alltext-nott`', '`allmath-nott`' [4, table 2]

2.5 (2013/03/13)

- Support for the `fontspec` package, viz. for OpenType fonts with LuaTeX and XeTeX
- Support for protrusion with XeTeX ≥ 0.9997

- Support for tracking/letterspacing with LuaTEX ≥ 0.62
- Allow context-sensitive setup with LuaTEX
- Info if protrusion settings are generic
- Protrusion settings for Latin Modern Roman (OpenType)
- Protrusion settings for Charis SIL (OpenType)
- Protrusion settings for Palatino Linotype (OpenType)

2.4 (2010/01/10)

- Protrusion settings for T2A encoded Minion

2.3e (2009/11/09)

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

2.3d (2009/03/27)

- New default for expansion option '`step`': 1, if pdfTEX ≥ 1.40 [3.3]

2.3c (2008/11/11)

- Support for LuaTEX enabled by default

2.3 (2007/12/23)

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

2.2 (2007/07/14)

- Improvements to tracking/letterspacing: retain kerning (pdfTEX ≥ 1.40.4); automatically adjust protrusion settings
- New key '`no ligatures`' for `\SetTracking` to disable selected or all ligatures (pdfTEX ≥ 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 ≥ 1.40.4) [5.2]
- New optional argument for `\DisableLigatures` to disable selected ligatures [8]
- New command `\DeclareMicrotypeVariants` to specify variant suffixes [5.7]
- New command `\textmicrotypecontext` as a wrapper for `\microtypecontext` [6]
- Protrusion settings for Bitstream Letter Gothic

2.1 (2007/01/21)

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

2.0 (2007/01/14)

- Support for the new extensions of pdfTEX ≥ 1.40: tracking/letterspacing, additional kerning, and adjustment of interword spacing (glue) (new commands `\SetTracking`, `\SetExtraKerning`, `\SetExtraSpacing`; new options '`tracking`', '`kerning`', '`spacing`') [5.3, 5.4, 5.5]
- New commands `\textls` and `\lsstyle` for letterspacing, new option '`letterspace`' [3.4, 7]
- New option '`babel`' for automatic micro-typographic adjustment to the selected language [3.5, 6]
- New font sets: '`smallcaps`', '`footnotesize`', '`scriptsize`' [4, table 2]

- New package ‘`letterspace`’ providing the commands for robust and hyphenatable letterspacing [7]

1.9e (2006/07/28)

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

1.9d (2006/05/05)

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

1.9c (2006/02/02)

- Protrusion settings for URW Garamond

1.9a (2005/12/05)

- Defer setup until the end of the preamble
- Inside the preamble, `\microtypesetup` accepts all package options [3.6]
- Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

- New command `\DisableLigatures` to disable ligatures (pdfTEX ≥ 1.30) [8]
- New command `\microtypecontext` to change the configuration context; new key ‘`context`’ for the configuration commands [6]
- New key ‘`font`’ to add single fonts to the font sets [4]
- New key ‘`preset`’ to set all characters to the specified value before loading the lists
- Value ‘`relative`’ renamed to ‘`character`’ for ‘`unit`’ keys
- Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
- Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)

1.8 (2005/06/23)

- New command `\DeclareMicrotypeSetDefault` to declare the default font sets [4]
- New option ‘`config`’ to load a different configuration file [3.5]
- New option ‘`unit`’ to measure protrusion factors relative to a dimension instead of the character width [5.1]
- Renamed commands from `\..MicroType..` to `\..Microtype..`
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The ‘`allmath`’ font set also includes U encoding
- Support for protrusion with the `ledmac` package (pdfTEX ≥ 1.30)

1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations [4, 5]
- New command `\LoadMicrotypeFile` to load a configuration file manually [5.7]
- New command `\Microtype@Hook` for font package authors [14.4.4]
- New option ‘`verbose=errors`’ to turn all warnings into errors
- Warning when running in draft mode

1.6 (2005/01/24)

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

- When pdfTEX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled [3.1]
- Use e-TEX extensions, if available

1.5 (2004/12/15)

- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled [3.1]
- New option ‘`selected`’ to enable selected expansion, default: `false` [3.3, 5.2]
- New default for expansion option ‘`step`’: 4 (min(`stretch`,`shrink`)/5) [3.3]
- Protrusion settings for Bitstream Charter

1.4 (2004/11/12)

- Set up fonts independently from 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`).

`pdftex-def`: Definitions specific to pdfTEX (`microtype-pdftex.def`).

`xetex-def`: Definitions specific to XeTEX (`microtype-xetex.def`).

`luatex-def`: Definitions specific to LuaTEX (`microtype-luatex.def`).

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

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

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

`luafile`: Lua functions (`microtype.lua`).

`config`: Surrounds all configuration modules.

- `cfg-t`: Surrounds (Latin) text configurations.
  - `m-t`: The main configuration file (`microtype.cfg`).
  - `bch`: Settings for Bitstream Charter (`mt-bch.cfg`).
  - `blg`: Settings for Bitstream Letter Gothic (`mt-blg.cfg`).
  - `cmr`: Settings for Computer Modern Roman (`mt-cmr.cfg`).
  - `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.

```
⟨*package|letterspace⟩
```

14.1 Preliminaries

`\MT@MT` This is us.

```
\def\MT@MT
⟨package⟩  {microtype}
⟨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.

```
\let\MT@restore@catcodes\@empty
```

```
\def\MT@fix@catcode#1#2{%
  \edef\MT@restore@catcodes{%
    \MT@restore@catcodes
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\MT@fix@catcode{17}{14}% ^^Q (comment)
\MT@fix@catcode{24} {9}% ^^X (ignore)
⟨package⟩\MT@fix@catcode{33}{12}% !
⟨package⟩\MT@fix@catcode{34}{12}% "
\MT@fix@catcode{36} {3}% $ (math shift)
\MT@fix@catcode{39}{12}% '
\MT@fix@catcode{42}{12}% *
\MT@fix@catcode{43}{12}% +
\MT@fix@catcode{44}{12}% ,
\MT@fix@catcode{45}{12}% -
\MT@fix@catcode{58}{12}% :
\MT@fix@catcode{60}{12}% <
\MT@fix@catcode{61}{12}% =
\MT@fix@catcode{62}{12}% >
⟨package⟩\MT@fix@catcode{63}{12}% ?
\MT@fix@catcode{94} {7}% ^ (superscript)
\MT@fix@catcode{96}{12}% `
⟨package⟩\MT@fix@catcode{124}{12}% |
```

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

```
⟨*package⟩
\newcommand*\DeclareMicrotypeSet[3][]{}
\newcommand*\UseMicrotypeSet[2][]{}
\newcommand*\DeclareMicrotypeSetDefault[2][]{}
\newcommand*\SetProtrusion[3][]{}
\newcommand*\SetExpansion[3][]{}
\newcommand*\SetTracking[3][]{}
\newcommand*\SetExtraKerning[3][]{}
\newcommand*\SetExtraSpacing[3][]{}
\newcommand*\DisableLigatures[2][]{}
\newcommand*\DeclareCharacterInheritance[3][]{}
\newcommand*\DeclareMicrotypeVariants[1]{}
\newcommand*\DeclareMicrotypeAlias[2]{}
\newcommand*\LoadMicrotypeFile[1]{}
\newcommand*\DeclareMicrotypeBabelHook[2]{}
\newcommand*\microtypesetup[1]{}
\newcommand*\microtypecontext[1]{}
\newcommand*\textmicrotypecontext[2]{#2}
\@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
⟨/package⟩
\newcommand*\lsstyle{}
\newcommand\textls[2][]{}
\def\textls#1#{}
\newcommand*\lslig[1]{#1}
⟨*package⟩
}
```

These commands also have a starred version.

```
\def\DeclareMicrotypeSet#1#{\@gobbletwo}
\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.

```
\@onlypreamble\DeclareMicrotypeSet
\@onlypreamble\UseMicrotypeSet
\@onlypreamble\DeclareMicrotypeSetDefault
\@onlypreamble\DisableLigatures
\@onlypreamble\DeclareMicrotypeVariants
\@onlypreamble\DeclareMicrotypeBabelHook
```

Don't load `letterspace`.

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

`\MT@old@cmd`

The old command names had one more hunch.

```
\def\MT@old@cmd#1#2{%
  \newcommand*#1{\MT@warning{%
    \string#1 is deprecated. Please use\MessageBreak
    \string#2 instead}%
    \let #1#2#2}}
```

```
\MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
\MT@old@cmd\DeclareMicroTypeSet  \DeclareMicrotypeSet
\MT@old@cmd\UseMicroTypeSet      \UseMicrotypeSet
\MT@old@cmd\LoadMicroTypeFile    \LoadMicrotypeFile
⟨/package⟩
```

`\MT@warning`
`\MT@warning@nl`
`\MT@info`
`\MT@info@nl`
`\MT@vinfo`
`\MT@error`
`\MT@warn@err`

Communicate.

```
\def\MT@warning{\PackageWarning\MT@MT}
\def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
⟨*package⟩
\def\MT@info{\PackageInfo\MT@MT}
\def\MT@info@nl#1{\MT@info{#1\@gobble}}
\let\MT@vinfo\@gobble
\def\MT@error{\PackageError\MT@MT}
\def\MT@warn@err#1{\MT@error{#1}{%
  This error message appears because you loaded the `\MT@MT'\MessageBreak
  package with the option `verbose=errors'. Consult the documentation\MessageBreak
  in \MT@MT.pdf to find out what went wrong.}}
```

14.1.1 Debugging

`\tracingmicrotype`
`\MT@dinfo`
`\MT@dinfo@nl`

Cases for `\tracingmicrotype`:

0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```
⟨*debug⟩
\MT@warning@nl{This is the debug version}
\newcount\tracingmicrotype
```

```
\tracingmicrotype=2
\def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
\def\MT@info@nl#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
\let\MT@vinfo\MT@info@nl
\def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
\def\MT@warning@nl#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
\def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
\def\MT@dinfo@nl#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@nl{#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

```
\newcount\tracingmicrotypeinpdf
```

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

```
\tracingmicrotypeinpdf=2
```

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

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

```
\RequirePackage{pdftexcmds}
\newif\ifMT@inannot \MT@inannottrue
\let\MT@pdf@annot\@empty
\def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
  {\def\MessageBreak{^^J\@spaces}%
   \MT@xadd\MT@pdf@annot{\pdf@escapestring{#1^^J}}}\fi\fi}
```

`\iftracingmicrotypeinpdfall`

With `\tracingmicrotypeinpdfallfalse`, 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.

```
\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).

```
\def\MT@show@pdfannot#1{%
  \ifnum\tracingmicrotypeinpdf<#1 \else
    \iftracingmicrotypeinpdfall\leavevmode\fi
    \pdfannot height 4pt width 4pt depth 2pt {%
      /Subtype/Caret
      /T(\expandafter\string\font@name)
      \ifcase#1\or
      /Subj(New font)/C[1 0 0]
      \else
      /Subj(Known font)/C[0 1 0]
      \fi
      /Contents(\MT@pdf@annot)
    }%
    \iftracingmicrotypeinpdfall\kern1pt \fi
    \global\MT@inannotfalse
  \fi
}
⟨/debug⟩
⟨/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`.

```
⟨*plain⟩
\def\MT@plain{2}
\ifx\documentclass\@undefined
  \def\MT@plain{1}
  \def\hmode@bgroup{\leavevmode\bgroup}
  \def\nfss@text#1{{\mbox{#1}}}
  \let\@typeset@protect\relax
  \ifx\eplain\@undefined
    \def\MT@plain{0}
    \def\PackageWarning#1#2{%
      \begingroup
        \newlinechar=10 %
        \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
        \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
      \endgroup
    }
    \def\on@line{ on input line \the\inputlineno}
    \def\@spaces{\space\space\space\space}
  \fi
\fi
```

`\MT@requires@latex`

Better use groups than plain ifs.

```
\def\MT@requires@latex#1{%
  \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
}
⟨/plain⟩
```

For definitions that depend on e-TeX features.

```
\ifcase 0%
  \ifx\eTeXversion\@undefined 1\else
    \ifx\eTeXversion\relax    1\else
      \ifcase\eTeXversion     1\fi
    \fi
  \fi
\else
  \catcode`\^^Q=9 \catcode`\^^X=14
\fi
⟨letterspace⟩^^Q\MT@warning@nl{This package requires the etex extensions.
⟨letterspace⟩^^Q               \MessageBreak Exiting}\MT@restore@catcodes\endinput
⟨debug⟩\MT@dinfo@nl{0}{this is
⟨debug⟩^^Q not
⟨debug⟩  etex}
```

We check whether we are running pdfTeX, XeTeX, or LuaTeX, and load the appropriate definition file.

`\MT@clear@options`

If we are using neither of these engines, we disable everything and exit.

```
\def\MT@clear@options{%
⟨plain⟩  \MT@requires@latex1{%
  \AtEndOfPackage{\let\@unprocessedoptions\relax\MT@restore@catcodes}%
  \let\CurrentOption\@empty
⟨package⟩  \let\MT@endinput\endinput
⟨plain⟩  }\relax
}
```

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.

```
\ifx\normalpdftexversion\@undefined \else
  \let\pdftexversion \normalpdftexversion
  \let\pdftexrevision\normalpdftexrevision
  \let\pdfoutput     \normalpdfoutput
\fi
```

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

```
\let\MT@engine\relax
⟨letterspace⟩\def\MT@engine@tooold{0}
\ifx\pdftexversion\@undefined \else
  \ifx\pdftexversion\relax \else
    \def\MT@engine{pdf}
⟨letterspace⟩    \let\MT@pdf@or@lua\@firstoftwo
⟨letterspace⟩    \ifnum\pdftexversion > 139 \def\MT@engine@tooold{1}\fi
  \fi
\fi
\ifx\directlua\@undefined \else
  \ifx\directlua\relax \else
    \def\MT@engine{lua}
```

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

```
⟨*letterspace⟩
    \let\MT@pdf@or@lua\@secondoftwo
    \ifnum\luatexversion < 62 \def\MT@engine@tooold{0}
    \else
      \def\MT@engine@tooold{1}
      \let\MT@lua\directlua
      \ifnum\luatexversion > 84
        \let\pdfoutput\outputmode
        \let\pdfprotrudechars\protrudechars
        \let\pdfadjustspacing\adjustspacing
      \fi
    \fi
⟨/letterspace⟩
  \fi
\fi
⟨*package⟩
\ifx\MT@engine\relax
  \ifx\XeTeXversion\@undefined \else
    \ifx\XeTeXversion\relax \else
      \def\MT@engine{xe}
    \fi
  \fi
\fi
⟨/package⟩
⟨/package|letterspace⟩
```

`\MT@pdftex@no` pdfTEX'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 pdfTEX 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 pdfTEX:

0: not running pdfTEX

1: pdfTEX (< 0.14f)

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

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

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

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

6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`[14]; `\pdftracingfonts`; always e-TEX (≥ 1.40)

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

```
⟨*pdftex-def⟩
⟨debug⟩\MT@dinfo@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
\def\MT@pdftex@no{7}
\ifnum\pdftexversion = 140
  \ifnum\pdftexrevision < 4
    \def\MT@pdftex@no{6}
  \fi
\else
  \ifnum\pdftexversion < 140
    \def\MT@pdftex@no{5}
    \ifnum\pdftexversion < 130
      \def\MT@pdftex@no{4}
      \ifnum\pdftexversion < 120
        \def\MT@pdftex@no{3}
        \ifnum\pdftexversion = 14
          \ifnum \expandafter`\pdftexrevision < `h
            \def\MT@pdftex@no{2}
            \ifnum \expandafter`\pdftexrevision < `f
              \def\MT@pdftex@no{1}
            \fi
          \fi
        \else
          \ifnum\pdftexversion < 14
            \def\MT@pdftex@no{1}
          \fi
        \fi
      \fi
    \fi
  \fi
\fi
⟨debug⟩\MT@dinfo@nl{0}{pdftex no.: \MT@pdftex@no}
⟨/pdftex-def⟩
```

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

```
⟨*xetex-def⟩
⟨debug⟩\MT@dinfo@nl{0}{this is xetex (\the\XeTeXversion\XeTeXrevision)}
\ifdim 0\XeTeXrevision pt < 0.9997pt
  \def\MT@xetex@no{1}
\else
  \def\MT@xetex@no{2}
\fi
⟨debug⟩\MT@dinfo@nl{0}{xetex no.: \MT@xetex@no}
⟨/xetex-def⟩
```

`\MT@luatex@no` Cases for LuaTEX (`\luatexversion` ought to have been enabled by the format):

0: N/A

1: LuaTEX (< 0.36)

2: + `\directlua` without state number (≥ 0.36)

---

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

3: + `\letterspacefont`; non-automatic expansion doesn't work anymore, and automatic expansion in DVI mode is realised by modifying the tracking, not the glyphs[15] (≥ 0.62)

4: + almost all of the pdfTEX primitives have been renamed (≥ 0.85)

5: + default `\efcode` = 1000; `\protrusionboundary` [not yet supported] (≥ 0.90)

Also, sometime between 1.0.4 and 1.0.7, the function `font.setexpansion` has been introduced, but we'll test this directly later.

```
⟨*luatex-def⟩
⟨debug⟩\MT@dinfo@nl0{this is luatex (\the\luatexversion)}
```

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

```
\let\MT@lua\directlua
\def\MT@luatex@no{5}
\ifnum\luatexversion<90
  \def\MT@luatex@no{4}
  \ifnum\luatexversion<85
    \def\MT@luatex@no{3}
    \ifnum\luatexversion<62
      \def\MT@luatex@no{2}
      \ifnum\luatexversion<36
        \def\MT@lua{\directlua0}
        \def\MT@luatex@no{1}
      \fi
    \fi
  \fi
\fi

⟨debug⟩\MT@dinfo@nl{0}{luatex no.: \MT@luatex@no}
⟨/luatex-def⟩

⟨*pdftex-def|xetex-def|letterspace⟩
\ifnum
⟨pdftex-def|xetex-def⟩  \csname MT@\MT@engine tex@no\endcsname < 2
⟨letterspace⟩  \MT@engine@tooold=\z@
  \MT@warning@nl{You
⟨*letterspace⟩
    \ifx\MT@engine\relax
      don't seem to be using pdftex or luatex.\MessageBreak
      Try running `pdftex' or `luatex' instead of\MessageBreak
        `\ifx\XeTeXversion\@undefined\else xe\fi tex'%
    \else
⟨/letterspace⟩
      are using a \MT@engine tex version older than
⟨pdftex-def⟩      0.14f%
⟨xetex-def⟩      0.9997%
⟨letterspace⟩        \MT@pdf@or@lua{1.40}{0.62}%
      .\MessageBreak
      `\MT@MT' does not work with this version.\MessageBreak
      Please install a newer version of \MT@engine tex%
⟨letterspace⟩    \fi
      .\MessageBreak I will quit now}
  \MT@clear@options
\endinput\fi
⟨/pdftex-def|xetex-def|letterspace⟩
```

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

```
⟨*package|letterspace⟩
```

---

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

```
\RequirePackage{keyval}[1997/11/10]
⟨*package⟩
```

`\MT@toks`

We need a token register.

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

`\ifMT@if@`

A scratch if.

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

14.1.3 Declarations

`\ifMT@protrusion`
`\ifMT@expansion`
`\ifMT@auto`
`\ifMT@selected`
`\ifMT@noligatures`
`\ifMT@draft`
`\ifMT@spacing`
`\ifMT@kerning`
`\ifMT@tracking`
`\ifMT@babel`

These are the global switches …

```
\newif\ifMT@protrusion
\newif\ifMT@expansion
\newif\ifMT@auto
\newif\ifMT@selected
\newif\ifMT@noligatures
\newif\ifMT@draft
\newif\ifMT@spacing
\newif\ifMT@kerning
\newif\ifMT@tracking
\newif\ifMT@babel
```

[This line intentionally left blank.]

`\MT@pr@level`
`\MT@ex@level`
`\MT@pr@factor`
`\MT@ex@factor`
`\MT@sp@factor`
`\MT@kn@factor`

… and numbers.

```
\let\MT@pr@level\tw@
\let\MT@ex@level\tw@
\let\MT@pr@factor\@m
\let\MT@ex@factor\@m
\let\MT@sp@factor\@m
\let\MT@kn@factor\@m
```

`\MT@pr@unit`
`\MT@sp@unit`
`\MT@kn@unit`

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

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

`\MT@stretch`
`\MT@shrink`
`\MT@step`

Expansion settings.

```
\let\MT@stretch\m@ne
\let\MT@shrink \m@ne
\let\MT@step   \m@ne
```

`\MT@pr@min`
`\MT@pr@max`
`\MT@ex@min`
`\MT@ex@max`
`\MT@sp@min`
`\MT@sp@max`
`\MT@kn@min`
`\MT@kn@max`
`\MT@tr@min`
`\MT@tr@max`

Minimum and maximum values allowed by pdfTEX.

```
\def\MT@pr@min{-\@m}
\let\MT@pr@max\@m
\let\MT@ex@min\z@
\let\MT@ex@max\@m
\def\MT@sp@min{-\@m}
\let\MT@sp@max\@m
\def\MT@kn@min{-\@m}
\let\MT@kn@max\@m
⟨/package⟩
\def\MT@tr@min{-\@m}
\let\MT@tr@max\@m
⟨*package⟩
```

`\MT@factor@default`

Default factor.

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

`\MT@stretch@default`
`\MT@shrink@default`

Default values for expansion.

```
\def\MT@stretch@default{20 }
```

```
\def\MT@shrink@default{20 }
```

\MT@letterspace
\MT@letterspace@default

Default value for letterspacing (in thousandths of 1 em).

```
⟨/package⟩
\let\MT@letterspace\m@ne
\def\MT@letterspace@default{100}
⟨*package⟩
```

\ifMT@document

Our private test whether we’re still in the preamble.

```
\newif\ifMT@document
⟨/package⟩
⟨/package|letterspace⟩
```

14.1.4 Auxiliary macros

\MT@requires@pdftex
\MT@requires@luatex

For definitions that depend on a particular pdfTEX resp. LuaTEX version.

```
⟨*pdftex-def|luatex-def⟩
\def
⟨pdftex-def⟩  \MT@requires@pdftex%
⟨luatex-def⟩  \MT@requires@luatex%
  #1{\ifnum
⟨pdftex-def⟩  \MT@pdftex@no
⟨luatex-def⟩  \MT@luatex@no
    <#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}
⟨luatex-def&debug⟩\MT@requires@luatex4{\MT@lua{tex.enableprimitives('pdf',{'tracingfonts'})}}\relax
⟨pdftex-def&debug⟩\MT@requires@pdftex6{
⟨debug⟩\pdftracingfonts=1
⟨pdftex-def&debug⟩}\relax
⟨/pdftex-def|luatex-def⟩
```

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

```
⟨*luatex-def⟩
\@ifl@t@r\fmtversion{2016/01/01}\relax{\RequirePackage{luatexbase}}
```

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

```
\@ifl@t@r\fmtversion{2017/01/01}\relax{\RequirePackage{luaotfload}}
\MT@lua{require("microtype")}
⟨/luatex-def⟩
```

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

```
⟨*luafile⟩

function microtype.info(...)
  luatexbase.module_info("microtype",...)
end

local find       = string.find
local match      = string.match
local tex_write  = tex.write

local catpackage
if luatexbase.registernumber then
  catpackage = luatexbase.registernumber("catcodetable@atletter") -- LaTeX
else
  catpackage = luatexbase.catcodetables.CatcodeTableLaTeXAtLetter -- luatexbase
end
function microtype.sprint (...)
```

```
  tex.sprint(catpackage, ...)
end

⟨/luafile⟩
```

To be continued, but first back to primitives.

`\MT@glet`

Here's the forgotten one.

```
⟨*package|letterspace⟩
\def\MT@glet{\global\let}
```

`\MT@exp@cs`
`\MT@exp@gcs`

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.

```
\def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
⟨*package⟩
\def\MT@exp@gcs#1#2{\begingroup\expandafter\endgroup\expandafter#1\csname#2\endcsname}
```

`\MT@def@n`
`\MT@gdef@n`

This is `\@namedef` and global.

```
\def\MT@def@n{\MT@exp@cs\def}
\def\MT@gdef@n{\MT@exp@gcs\gdef}
```

`\MT@edef@n`
`\MT@xdef@n`

Its expanding versions.

```
⟨/package⟩
\def\MT@edef@n{\MT@exp@cs\edef}
⟨*package⟩
\def\MT@xdef@n{\MT@exp@gcs\xdef}
```

`\MT@let@nc`
`\MT@glet@nc`

`\let` a `\csname` sequence to a command.

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

`\MT@let@cn`

`\let` a command to a `\csname` sequence.

```
⟨/package⟩
\def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
⟨*package⟩
```

`\MT@let@nn`
`\MT@glet@nn`

`\let` a `\csname` sequence to a `\csname` sequence.

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

`\MT@@font`

Remove trailing space from the font name.

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

`\MT@exp@one@n`

Expand the second token once and enclose it in braces.

```
⟨/package⟩
\def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}
```

`\MT@exp@two@c`

Expand the next two tokens after ⟨*#1*⟩ once.

```
\def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
⟨*package⟩
```

`\MT@exp@two@n`

Expand the next two tokens after ⟨*#1*⟩ once and enclose them in braces.

```
\def\MT@exp@two@n#1#2#3{%
  \expandafter\expandafter\expandafter
    #1\expandafter\expandafter\expandafter
      {\expandafter#2\expandafter}\expandafter{#3}}
```

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

`\MT@ifdefined@c@T`
`\MT@ifdefined@c@TF`
`\MT@ifdefined@n@T`
`\MT@ifdefined@n@TF`

Wrapper for testing whether command resp. `\csname` sequence is defined. If we are running e-TEX, we will use its primitives `\ifdefined` and `\ifcsname`, which decreases memory use substantially.

```
\def\MT@ifdefined@c@T#1{%
^^X  \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
^^Q  \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
}
⟨/package⟩
\def\MT@ifdefined@c@TF#1{%
^^X  \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
⟨package⟩^^Q  \ifx#1\@undefined
⟨package⟩^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
}
\def\MT@ifdefined@n@T#1{%
^^X  \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
⟨package⟩^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
⟨package⟩^^Q    \expandafter\@gobble\else\expandafter\@firstofone\fi
}
\def\MT@ifdefined@n@TF#1{%
^^X  \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
⟨package⟩^^Q  \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
⟨package⟩^^Q    \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
}
⟨*package⟩
```

`\MT@detokenize@n`
`\MT@detokenize@c`
`\MT@rem@last@space`

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).

```
\def\MT@detokenize@n#1{%
^^X  \expandafter\MT@rem@last@space\detokenize{#1} \@nil
^^Q  \string#1%
}
\def\MT@detokenize@c#1{%
^^X  \MT@exp@one@n\MT@detokenize@n#1%
^^Q  \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
}
\def\MT@rem@last@space#1 #2{#1%
  \ifx\@nil#2\else \space
  \expandafter\MT@rem@last@space\expandafter#2\fi
}
```

`\MT@ifempty`

Test whether argument is empty.

```
⟨/package⟩
\begingroup
\catcode`\%=12
\catcode`\&=14
\gdef\MT@ifempty#1{&
  \if %#1%&
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
\endgroup
⟨*package⟩
```

`\MT@ifint`

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

```
⟨/package⟩
⟨/package|letterspace⟩
⟨pdftex-def⟩\MT@requires@pdftex6{
⟨letterspace⟩\MT@pdf@or@lua{
⟨*pdftex-def|letterspace⟩
\def\MT@ifint#1{%
  \ifcase\pdfmatch{^-*[0-9]+ *$}{#1}\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi
}
}{
⟨/pdftex-def|letterspace⟩
⟨*pdftex-def|xetex-def|letterspace⟩
\def\MT@ifint#1{%
  \if!\ifnum9<1#1!\else?\fi
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
⟨/pdftex-def|xetex-def|letterspace⟩
⟨pdftex-def|letterspace⟩}
⟨luatex-def⟩\def\MT@ifint#1{\csname\MT@lua{microtype.if_int([[#1]])}\endcsname}
⟨*luafile⟩
local function if_int(s)
  if find(s,"^-*[0-9]+ *$") then
    tex_write("@firstoftwo")
  else
    tex_write("@secondoftwo")
  end
end
microtype.if_int = if_int

⟨/luafile⟩
```

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

```
⟨*pdftex-def⟩
\MT@requires@pdftex6{
\def\MT@ifdimen#1{%
  \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
                    (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi
}
}{
⟨/pdftex-def⟩
⟨*pdftex-def|xetex-def⟩
\def\MT@ifdimen#1{%
  \setbox\z@=\hbox{%
    \MT@count=1#1\relax
    \ifnum\MT@count=\@ne
      \aftergroup\@secondoftwo
    \else
      \aftergroup\@firstoftwo
    \fi
  }%
}
⟨/pdftex-def|xetex-def⟩
⟨pdftex-def⟩}
⟨luatex-def⟩\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen([[#1]])}\endcsname}
⟨*luafile⟩
local function if_dimen(s)
  if (find(s, "^-*[0-9]+(%a*) *$") or
      find(s, "^-*[0-9]*[.,][0-9]+(%a*) *$")) then
    tex_write("@firstoftwo")
  else
    tex_write("@secondoftwo")
  end
end
microtype.if_dimen = if_dimen
```

```

⟨/luafile⟩
```

\MT@ifdim

Compare floating point numbers.

```
⟨*package⟩
\def\MT@ifdim#1#2#3{%
  \ifdim #1\p@ #2 #3\p@
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
⟨/package⟩
```

\MT@ifstreq

Test whether two strings (fully expanded) are equal.

```
⟨*pdftex-def|xetex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex5{
\def\MT@ifstreq#1#2{%
⟨pdftex-def⟩  \ifnum\pdfstrcmp{#1}{#2}=\z@
⟨xetex-def⟩  \ifnum\strcmp{#1}{#2}=\z@
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
⟨/pdftex-def|xetex-def⟩
⟨*pdftex-def⟩
}{
\def\MT@ifstreq#1#2{%
  \edef\MT@res@a{#1}%
  \edef\MT@res@b{#2}%
  \ifx\MT@res@a\MT@res@b
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
}
⟨/pdftex-def⟩
⟨luatex-def⟩\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq([[#1]],[[#2]])}\endcsname}
⟨*luafile⟩
local function if_str_eq(s1, s2)
  if s1 == s2 then
    tex_write("@firstoftwo")
  else
    tex_write("@secondoftwo")
  end
end
microtype.if_str_eq = if_str_eq

⟨/luafile⟩
```

\MT@xadd

Add item to a list.

```
⟨*package⟩
\def\MT@xadd#1#2{%
  \ifx#1\relax
    \xdef#1{#2}%
  \else
    \xdef#1{#1#2}%
  \fi
}
```

\MT@xaddb

Add item to the beginning.

```
\def\MT@xaddb#1#2{%
  \ifx#1\relax
```

```
    \xdef#1{#2}%
  \else
    \xdef#1{#2#1}%
  \fi
}
⟨/package⟩
```

`\MT@map@clist@n`
`\MT@map@clist@c`
`\MT@map@clist@`
`\MT@clist@function`
`\MT@clist@break`

Run ⟨*#2*⟩ on all elements of the comma list ⟨*#1*⟩. This and the following is modelled after LaTeX3 commands.

```
⟨*package|letterspace⟩
\def\MT@map@clist@n#1#2{%
  \ifx\@empty#1\else
    \def\MT@clist@function##1{#2}%
    \MT@map@clist@#1,\@nil,\@nnil
  \fi
}

\def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
\def\MT@map@clist@#1,{%
  \ifx\@nil#1%
    \expandafter\MT@clist@break
  \fi
  \MT@clist@function{#1}%
  \MT@map@clist@
}
\let\MT@clist@function\@gobble
\def\MT@clist@break#1\@nnil{}
⟨*package⟩
```

`\MT@map@tlist@n`
`\MT@map@tlist@c`
`\MT@map@tlist@`
`\MT@tlist@break`

Execute ⟨*#2*⟩ on all elements of the token list ⟨*#1*⟩. `\MT@tlist@break` can be used to jump out of the loop.

```
\def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
\def\MT@map@tlist@#1#2{%
  \ifx\@nnil#2\else
    #1{#2}%
    \expandafter\MT@map@tlist@
    \expandafter#1%
  \fi
}
\def\MT@tlist@break#1\@nnil{\fi}
```

`\ifMT@inlist@`
`\MT@in@clist`

Test whether item ⟨*#1*⟩ is in comma list ⟨*#2*⟩. Using `\pdfmatch` would be slower.

```
\newif\ifMT@inlist@
\def\MT@in@clist#1#2{%
  \def\MT@res@a##1,#1,##2##3\@nnil{%
    \ifx##2\@empty
      \MT@inlist@false
    \else
      \MT@inlist@true
    \fi
  }%
  \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
}
```

`\MT@rem@from@clist`

Remove item ⟨*#1*⟩ from comma list ⟨*#2*⟩. This is basically `\@removeelement` from `ltcntrl.dtx`. Using `\pdfmatch` and `\pdflastmatch` here would be really slow!

```
\def\MT@rem@from@clist#1#2{%
  \def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
  \def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
  \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
}
```

`\MT@in@tlist`
`\MT@in@tlist@`

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

```
\def\MT@in@tlist#1#2{%
  \MT@inlist@false
  \def\MT@res@a{#1}%
  \MT@map@tlist@c#2\MT@in@tlist@
}
\def\MT@in@tlist@#1{%
  \edef\MT@res@b{#1}%
  \ifx\MT@res@a\MT@res@b
    \MT@inlist@true
    \expandafter\MT@tlist@break
  \fi
}
```

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

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

```
\def\MT@in@rlist#1{%
  \MT@inlist@false
  \MT@map@tlist@c#1\MT@in@rlist@
}
\def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
\def\MT@in@rlist@@#1#2#3{%
  \MT@ifdim{#2}=\m@ne{%
    \MT@ifdim{#1}=\MT@size
      \MT@inlist@true
      \relax
  }{%
    \MT@ifdim\MT@size<{#1}\relax{%
      \MT@ifdim\MT@size<{#2}%
        \MT@inlist@true
        \relax
    }%
  }%
  \ifMT@inlist@
    \def\MT@size@name{#3}%
    \expandafter\MT@tlist@break
  \fi
}
```

`\MT@loop`
`\MT@iterate`
`\MT@repeat`

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

```
⟨/package⟩
\def\MT@loop#1\MT@repeat{%
  \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
  \MT@iterate \let\MT@iterate\relax
}
\let\MT@repeat\fi
```

`\MT@while@num`

Execute ⟨*#3*⟩ from ⟨*#1*⟩ up to (excluding) ⟨*#2*⟩ (much faster than LaTeX's `\@whilenum`).

```
\def\MT@while@num#1#2#3{%
  \@tempcnta#1\relax
  \MT@loop #3%
    \advance\@tempcnta \@ne
    \ifnum\@tempcnta < #2\MT@repeat
}
⟨/package|letterspace⟩
```

`\MT@if@luaotf@font`

For fonts loaded by `luaotfload` we query the font's table.

```
⟨letterspace⟩\MT@pdf@or@lua{\let\MT@if@luaotf@font\@secondoftwo}{
⟨luatex-def|letterspace⟩\def\MT@if@luaotf@font{\csname\MT@lua{%
⟨luatex-def⟩  microtype.if_luaotf_font()
⟨*luafile|letterspace⟩
⟨luafile⟩local function if_luaotf_font()
  local thefont = font.getfont(font.current())
  if thefont and ( thefont.format == "opentype" or thefont.format == "truetype" )
    then tex.write("@firstoftwo")
    else tex.write("@secondoftwo")
  end
⟨luafile⟩end
⟨luafile⟩microtype.if_luaotf_font = if_luaotf_font
⟨luafile⟩
⟨/luafile|letterspace⟩
⟨luatex-def|letterspace⟩ }\endcsname
⟨luatex-def|letterspace⟩}
⟨letterspace⟩}
```

`\MT@do@font`

Execute ⟨*#1*⟩ 256 times,

```
⟨pdftex-def|letterspace⟩\def\MT@do@font{\MT@while@num\z@\@cclvi}
```

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

```
⟨*luatex-def⟩
\def\MT@do@font#1{%
  \MT@if@luaotf@font{%
    \def\MT@dofont@function{#1}%
    \MT@lua{microtype.do_font()}%
  }{\MT@while@num\z@\@cclvi{#1}}%
}
⟨/luatex-def⟩
```

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

```
⟨*luafile⟩
local function do_font()
  local thefont = font.getfont(font.current())
  if thefont then
    for i,v in next,thefont.characters do
      if v.index == nil or v.index > 0 then
        microtype.sprint([[\@tempcnta=]]..i..[[\relax\MT@dofont@function]])
      end
    end
  end
end
microtype.do_font = do_font

⟨/luafile⟩
```

The XƎTEX variant (it's slow . . . !).

```
⟨*xetex-def⟩
\def\MT@do@font#1{%
  \@tempcnta=\z@
  \MT@loop
    \iffontchar\MT@font\@tempcnta #1\fi
    \advance\@tempcnta\@ne
    \ifnum\@tempcnta < \XeTeXlastfontchar\MT@font \MT@repeat
}
⟨/xetex-def⟩
⟨*package⟩
```

`\MT@count`
`\MT@increment`

Increment macro ⟨*#1*⟩ by one. Saves using up too many counters. The e-TEX way is slightly faster.

```
\newcount\MT@count
\def\MT@increment#1{%
^^X  \edef#1{\number\numexpr #1 + 1\relax}%
^^Q  \MT@count=#1\relax
^^Q  \advance\MT@count \@ne
^^Q  \edef#1{\number\MT@count}%
}
```

`\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.

```
\def\MT@scale#1#2#3{%
^^Q  \multiply #1 #2\relax
  \ifnum #3 = \z@
^^X    #1=\numexpr #1 * #2\relax
  \else
^^X    #1=\numexpr #1 * #2 / #3\relax
^^Q    \divide #1 #3\relax
  \fi
}
```

`\MT@abbr@pr`
`\MT@abbr@ex`
`\MT@abbr@pr@c`
`\MT@abbr@ex@c`
`\MT@abbr@pr@inh`
`\MT@abbr@ex@inh`
`\MT@abbr@nl`
`\MT@abbr@sp`
`\MT@abbr@sp@c`
`\MT@abbr@sp@inh`
`\MT@abbr@kn`
`\MT@abbr@kn@c`
`\MT@abbr@kn@inh`
`\MT@abbr@tr`
`\MT@abbr@tr@c`

Some abbreviations. Thus, we can have short command names but full-length log output.

```
\def\MT@abbr@pr{protrusion}
\def\MT@abbr@ex{expansion}
\def\MT@abbr@pr@c{protrusion codes}
\def\MT@abbr@ex@c{expansion codes}
\def\MT@abbr@pr@inh{protrusion inheritance}
\def\MT@abbr@ex@inh{expansion inheritance}
\def\MT@abbr@nl{noligatures}
\def\MT@abbr@sp{spacing}
\def\MT@abbr@sp@c{interword spacing codes}
\def\MT@abbr@sp@inh{interword spacing inheritance}
\def\MT@abbr@kn{kerning}
\def\MT@abbr@kn@c{kerning codes}
\def\MT@abbr@kn@inh{kerning inheritance}
\def\MT@abbr@tr{tracking}
\def\MT@abbr@tr@c{tracking amount}
```

`\MT@rbba@protrusion`
`\MT@rbba@expansion`
`\MT@rbba@spacing`
`\MT@rbba@kerning`
`\MT@rbba@tracking`

These we also need the other way round.

```
\def\MT@rbba@protrusion{pr}
\def\MT@rbba@expansion{ex}
\def\MT@rbba@spacing{sp}
\def\MT@rbba@kerning{kn}
\def\MT@rbba@tracking{tr}
```

`\MT@features`
`\MT@features@long`

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

```
\def\MT@features{pr,ex,sp,kn,tr}
\def\MT@features@long{protrusion,expansion,spacing,kerning,tracking}
```

`\MT@is@feature`

Whenever an optional argument accepts a list of features, we can use this command to check whether a feature exists in order to prevent a rather confusing '`Missing \endcsname inserted`' error message. The feature (long form) must be in ⟨*#1*⟩, the type of list to ignore in ⟨*#2*⟩, then comes the action.

```
\def\MT@is@feature#1#2{%
  \MT@in@clist{#1}\MT@features@long
  \ifMT@inlist@
    \expandafter\@firstofone
  \else
    \MT@error{`#1' is not an available micro-typographic\MessageBreak
      feature. Ignoring #2}{Available features are: `\MT@features@long'.}%
    \expandafter\@gobble
  \fi
}
```

14.1.5 Compatibility

For the record, the following LATEX kernel commands will be modified by `microtype`:

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent` (all in section 14.2.9)
- `\showhyphens` (in section 14.4.6)

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

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

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

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

```
⟨/package⟩
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\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.

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

Don't hesitate with `miniltx`.

```
⟨plain⟩}{\let\MT@addto@setup\@firstofone}
```

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

```
\def\MT@with@package@T#1{\@ifpackageloaded{#1}\@firstofone\@gobble}
⟨/package|letterspace⟩
⟨*package⟩
```

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

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

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

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

```
⟨/package⟩
⟨pdftex-def⟩\MT@requires@pdftex5{
⟨*pdftex-def|luatex-def|xetex-def⟩
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@ifdefined@c@TF\l@dunhbox@line{%
```

`\MT@led@unhbox@line`

Hook.

```
        \MT@info@nl{Patching ((r)e)ledmac to enable character protrusion}%
        \let\MT@led@unhbox@line\l@dunhbox@line
        \renewcommand*{\l@dunhbox@line}[1]{%
          \ifhbox##1%
            \kern\leftmarginkern##1%
            \expandafter\MT@led@unhbox@line\expandafter##1\expandafter
            \kern\rightmarginkern##1%
          \fi
        }%
```

```
      }{%
        \MT@warning@nl{%
          Character protrusion in paragraphs with line\MessageBreak
          numbering will only work if you update ledmac,\MessageBreak
          or use one of its successors, eledmac or reledmac}%
      }%
    \fi
  }
⟨/pdftex-def|luatex-def|xetex-def⟩
⟨*pdftex-def⟩
}{
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@warning@nl{%
        The pdftex version you are using does not allow\MessageBreak
        character protrusion in paragraphs with line\MessageBreak
        numbering by the `((r)e)ledmac' package.\MessageBreak
        Upgrade pdftex to version 1.30 or later}%
    \fi
  }
}
⟨/pdftex-def⟩
```

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

`\MT@restore@p@h`

Restore meaning of \% and \#.

```
⟨*package|letterspace⟩
⟨*package⟩
\def\MT@restore@p@h{\chardef\%`\% \chardef\#`\# }
```

`\ifMT@xunicode`
`\ifMT@fontspec`

Two new conditionals for use with XƎTEX or LuaTEX.

```
\newif\ifMT@xunicode
\MT@with@package@T{xunicode}\MT@xunicodetrue
\newif\ifMT@fontspec
\MT@with@package@T{fontspec}\MT@fontspectrue
```

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

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

```
\let\MT@maybe@gobble@with@tikz\@firstofone
\def\MT@tikz@setup{%
  \def\MT@maybe@gobble@with@tikz{%
    \ifnum\tikz@expandcount>\z@
```

```
      \expandafter\@gobble
    \else
      \expandafter\@firstofone
    \fi}}
```

\MT@setupfont@hook

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

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

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

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

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

```
  \MT@if@false
  \MT@with@babel@and@T{spanish} \MT@if@true
  \MT@with@babel@and@T{galician}\MT@if@true
  \MT@with@babel@and@T{mexican} \MT@if@true
  \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.)

```
  \MT@with@package@T{csquotes}{%
    \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax}%
```

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

```
  \MT@if@false
  \MT@with@package@T{hyperref}  \MT@if@true
  \MT@with@package@T{tex4ht}    \MT@if@true
  \MT@with@package@T{mathastext}\MT@if@true
  \ifMT@if@\MT@restore@p@h\fi
  \MT@with@package@T{tikz}\MT@tikz@setup
}
```

Check again at the end of the preamble.

```
⟨/package⟩
\MT@addto@setup{%
⟨*package⟩
```

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

```
  \MT@with@package@T{pdfcprot}{%
    \MT@error{Detected the `pdfcprot' package!\MessageBreak
              `\MT@MT' and `pdfcprot' may not be used together}{%
The `pdfcprot' package provides an interface to character protrusion.\MessageBreak
So does the `\MT@MT' package. Using both packages at the same\MessageBreak
time will almost certainly lead to undesired results. Have your choice!}%
  }%
  \MT@with@package@T  {ledmac}\MT@ledmac@setup
  \MT@with@package@T {eledmac}\MT@ledmac@setup
  \MT@with@package@T{reledmac}\MT@ledmac@setup
  \MT@with@package@T{xunicode}\MT@xunicodetrue
  \MT@with@package@T{fontspec}\MT@fontspectrue
```

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

```
  \MT@glet\MT@setupfont@hook\@empty
  \MT@if@false
  \MT@with@babel@and@T{spanish} \MT@if@true
  \MT@with@babel@and@T{galician}\MT@if@true
  \MT@with@babel@and@T{mexican} \MT@if@true
  \ifMT@if@
    \g@addto@macro\MT@setupfont@hook{%
      \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
  \fi
  \MT@with@package@T{csquotes}{%
    \@ifpackagelater{csquotes}{2005/05/11}{%
      \g@addto@macro\MT@setupfont@hook\@disablequotes
    }{%
      \MT@warning@nl{%
        Should you receive warnings about unknown slot\MessageBreak
        numbers, try upgrading the `csquotes' package}%
    }%
  }%
```

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

```
  \MT@if@false
⟨/package⟩
⟨plain⟩  \MT@requires@latex2{
  \MT@with@package@T{hyperref}{%
    \pdfstringdefDisableCommands{%
⟨*package⟩
      \MT@ltx@pickupfont
      \let\textmicrotypecontext\@secondoftwo
      \let\microtypecontext\@gobble
⟨/package⟩
      \def\lsstyle{\pdfstringdefWarn\lsstyle}%
      \def\textls#1#{\pdfstringdefWarn\textls}%
    }%
⟨package⟩    \MT@if@true
  }%
⟨plain⟩  }\relax
⟨*package⟩
  \MT@with@package@T{tex4ht}\MT@if@true
  \MT@with@package@T{mathastext}\MT@if@true
  \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
```

The `listings` package makes numbers and letters active,

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

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

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

Inside a listing, `\space` is redefined.

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

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

```
      \let\lst@ProcessLetter\@empty
    }%
  }%
```

Of course, using both `soul`'s and `microtype`'s letterspacing mechanisms at the same time doesn't make much sense. But `soul` can do more, e.g., underlining. The optional argument to `\textls` may not be used. Also, we have to disable expansion within `soul`'s trial run. Under plain TEX, `soul` doesn't register itself the LATEX way, so we just test for its main command.

```
⟨/package⟩
  \ifx\SOUL@\@undefined\else
    \soulregister\lsstyle 0%
    \soulregister\textls  1%
    \ifx\XeTeXrevision\@undefined
      \let\MT@SOUL@doword\SOUL@doword
      \def\SOUL@doword{\pdfadjustspacing=\z@ \MT@SOUL@doword}%
    \fi
  \fi
⟨*package⟩
  \MT@with@package@T{tikz}\MT@tikz@setup
```

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

```
  \MT@with@package@T{pinyin}{%
    \let\MT@orig@py@macron\py@macron
    \@ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
      \def\py@macron#1#2{%
        \MT@ltx@pickupfont
        \MT@orig@py@macron{#1}{#2}%
        \MT@MT@pickupfont}%
    }{%
      \def\py@macron#1{%
        \MT@ltx@pickupfont
        \MT@orig@py@macron{#1}%
        \MT@MT@pickupfont}%
    }%
  }%
⟨/package⟩
}
⟨/package|letterspace⟩
```

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

```
⟨package⟩\expandafter\ifx\the\font\nullfont\normalfont\fi
```

14.2 Font setup

`\MT@setupfont`

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

```
⟨*pdftex-def|xetex-def|luatex-def⟩
\def\MT@setupfont{%
```

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

```
⟨xetex-def|luatex-def⟩  \MT@font
```

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

```
  \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).

```
⟨pdftex-def⟩\MT@requires@pdftex7{
⟨pdftex-def|luatex-def⟩\g@addto@macro\MT@setupfont\MT@copy@font
⟨pdftex-def⟩}\relax
```

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

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

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

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

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

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

```
⟨pdftex-def⟩\MT@requires@pdftex6
⟨luatex-def⟩\MT@requires@luatex3
⟨pdftex-def|luatex-def⟩  {\g@addto@macro\MT@setupfont\MT@tracking}\relax
\g@addto@macro\MT@setupfont{%
  \MT@check@font
  \ifMT@inlist@
⟨debug⟩\MT@show@pdfannot2%
  \else
    \MT@vinfo{Setting up font `\MT@@font'\on@line}%
    \MT@info@notracking
```

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.

```
  \MT@protrusion
⟨pdftex-def|luatex-def⟩  \MT@expansion
}
```

Interword spacing and kerning (pdfTEX 1.40).

```
⟨*pdftex-def⟩
\MT@requires@pdftex6{
\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}
}\relax
⟨/pdftex-def⟩
```

Disable ligatures (pdfTEX 1.30).

```
⟨pdftex-def⟩\MT@requires@pdftex5{
⟨pdftex-def|luatex-def⟩\g@addto@macro\MT@setupfont\MT@noligatures
⟨pdftex-def⟩}\relax
\g@addto@macro\MT@setupfont{%
```

Debugging.

```
⟨debug⟩\MT@show@pdfannot1%
```

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

```
    \MT@register@font
  \fi
}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

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

```
⟨*pdftex-def|luatex-def⟩
\let\MT@copy@font\relax
⟨pdftex-def⟩\MT@requires@pdftex7{
\def\MT@copy@font@{%
```

`\MT@font@copy`

For every new protrusion and expansion context, we create a new copy.

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

`\MT@font@orig`

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

```
    \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
    \expandafter\ifx\MT@font@orig\relax
      \MT@exp@two@c\MT@glet\MT@font@orig\font@name
    \else
      \MT@exp@two@c\let\font@name\MT@font@orig
    \fi
⟨pdftex-def⟩    \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
```

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

```
⟨luatex-def⟩    \MT@exp@two@c\MT@lua@copyfont\meaning\font@name\@nil
⟨debug⟩\MT@dinfo1{creating new copy: \MT@font@copy}%
```

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

```
    \MT@map@clist@c\MT@active@features{%
      \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
        \def\@tempa{##1}%
        \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@rem@from@list
      \fi
    }%
  \fi
  \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

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

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

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

`\MT@rem@from@list`

```
\def\MT@rem@from@list#1{%
  \MT@exp@cs\ifx{MT@\@tempa @#1font@list}\relax\else
    \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
       \MT@font \csname MT@\@tempa @#1font@list\endcsname
  \fi
}
⟨pdftex-def⟩}\relax
```

`\MT@lua@copy@font`

⟨*#1*⟩ and ⟨*#2*⟩ are '`select`' and '`font`', respectively, ⟨*#3*⟩ is the font spec.

```
⟨luatex-def⟩\def\MT@lua@copyfont #1 #2 #3\@nil{%
⟨luatex-def⟩  \global\expandafter\font\MT@font@copy=#3\relax}
⟨/pdftex-def|luatex-def⟩
```

Here's the promised dirty trick for users of older pdfTEX versions, which works around the problem that the use of the same font with different expansion parameters is prohibited. If you do not want to create a clone of the font setup (this would require duplicating the `tfm`/`vf` files under a new name, and writing new `fd` files and `map` entries), you can load a minimally larger font for the paragraph in question. E.g., for a document typeset in 10 pt:

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

Note that the `\expandpar` command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the `fix-cm` package to be able to select fonts in arbitrary sizes. Finally, the reason I suggest to use a larger font, and not a smaller one, is to prevent a different design size being selected.

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

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

```
⟨*pdftex-def|luatex-def|xetex-def⟩
\def\MT@fix@fontdimen@six{%
  \ifnum\fontdimen6\MT@font=\z@
    \fontdimen6\MT@font=%
⟨pdftex-def⟩      \pdffontsize\MT@font
⟨luatex-def⟩      \MT@requires@luatex4{\pdffeedback fontsize}{\pdffontsize}\MT@font
⟨xetex-def⟩      \MT@size pt
    \MT@info{Fixing zero \string\fontdimen 6 for font `\MT@@font'\MessageBreak
             (new value: \the\fontdimen6\MT@font)}%
    \MT@glet@nc{\MT@@font-fake6}\@empty
  \fi
  \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
}
⟨/pdftex-def|luatex-def|xetex-def⟩
```

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

Split up the font name (⟨*#6*⟩ may be a protrusion/expansion context and/or a letterspacing amount). With `fontspec` we also need to remove its internal instance counter.

```
⟨*package⟩
\def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
  \def\MT@encoding{#1}%
  \ifMT@fontspec
    \edef\MT@family{\MT@scrubfeature#2()\relax}%
  \else
    \def\MT@family{#2}%
  \fi
  \def\MT@series  {#3}%
  \def\MT@shape   {#4}%
  \def\MT@size    {#5}%
  \MT@fix@fontdimen@six
```

`\MT@familyalias`

Alias family?

```
  \MT@ifdefined@n@TF{MT@\MT@family @alias}%
    {\MT@let@cn\MT@familyalias{MT@\MT@family @alias}}%
    {\let\MT@familyalias\@empty}%
}
```

`\MT@scrubfeature`
`\MT@scrubfeatures`

Remove one resp. all feature counters (`fontspec`).

```
\def\MT@scrubfeature#1(#2)#3\relax{#1}
\def\MT@scrubfeatures#1(#2)#3\relax{%
  #1%
  \ifx\relax#3\relax\else
    \MT@scrubfeatures#3\relax
  \fi
}
```

`\ifMT@do`
`\MT@feat`
`\MT@maybe@do`

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

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

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

```
  \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`).

```
    \MT@dotrue
    \edef\@tempa{\csname MT@#1@setname\endcsname}%
    \MT@map@clist@n{font,encoding,family,series,shape,size}{%
      \MT@ifdefined@n@TF{MT@checklist@##1}%
        {\csname MT@checklist@##1\endcsname}%
        {\MT@checklist@{##1}}%
      {#1}%
    }%
  \else
    \MT@dofalse
  \fi
  \ifMT@do
```

`\MT@feat` stores the current feature.

```
    \def\MT@feat{#1}%
    \csname MT@set@#1@codes\endcsname
  \else
    \MT@ifstreq{#1}{tr}%
      {\let\MT@info@notracking\MT@info@notracking@}%
      {\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}%
  \fi
}
```

`\MT@info@notracking`
`\MT@info@notracking@`

To defer the message to after the font has actually been logged.

```
\let\MT@info@notracking\relax
\def\MT@info@notracking@{\MT@vinfo{... No tracking}}
```

`\MT@dinfo@list`

```
⟨debug⟩\def\MT@dinfo@list#1#2#3{\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: #2
⟨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).

```
\def\MT@checklist@#1#2{%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#2list@#1@\@tempa}{%
```

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

```
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist
      \csname MT@#1\expandafter\endcsname
      \csname MT@#2list@#1@\@tempa\endcsname
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#2}{#1}{in}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@list{#2}{#1}{not in}%
      \MT@dofalse
      \expandafter\MT@clist@break
    \fi
  }%
```

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.

```
⟨debug⟩  {\MT@dinfo@list{#2}{#1}{}}%
}
```

\MT@checklist@family

Also test for the alias font, if the original font is not in the list.

```
\def\MT@checklist@family#1{%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#1list@family@\@tempa}{%
    \MT@exp@two@n\MT@in@clist
        \MT@family{\csname MT@#1list@family@\@tempa\endcsname}%
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#1}{family}{in}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@list{#1}{family}{not in}%
      \MT@dofalse
      \ifx\MT@familyalias\@empty \else
        \MT@exp@two@n\MT@in@clist
            \MT@familyalias{\csname MT@#1list@family@\@tempa\endcsname}%
        \ifMT@inlist@
⟨debug⟩  \MT@dinfo@list{#1}{family alias}{in}%
          \MT@dotrue
⟨debug⟩\else\MT@dinfo@list{#1}{family alias}{not in}%
        \fi
      \fi
    \fi
    \ifMT@do \else
      \expandafter\MT@clist@break
    \fi
  }%
⟨debug⟩  {\MT@dinfo@list{#1}{family}{}}%
}
```

\MT@checklist@size

Test whether font size is in list of size ranges.

```
\def\MT@checklist@size#1{%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#1list@size@\@tempa}{%
    \MT@exp@cs\MT@in@rlist{MT@#1list@size@\@tempa}%
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#1}{size}{in}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@list{#1}{size}{not in}%
      \MT@dofalse
      \expandafter\MT@clist@break
    \fi
  }%
⟨debug⟩  {\MT@dinfo@list{#1}{size}{}}%
}
```

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

```
\def\MT@checklist@font#1{%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#1list@font@\@tempa}{%
```

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

```
    \edef\@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
      \@tempb \csname MT@#1list@font@\@tempa\endcsname
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@list{#1}{font}{in}%
      \expandafter\MT@clist@break
    \else
⟨debug⟩\MT@dinfo@list{#1}{font}{not in}%
      \MT@dofalse
    \fi
  }%
⟨debug⟩  {\MT@dinfo@list{#1}{font}{}}%
}
```

14.2.1 Protrusion

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

```
\newif\ifMT@nofamily
⟨/package⟩
```

`\MT@protrusion` Set up for protrusion?

```
⟨*pdftex-def|xetex-def|luatex-def⟩
\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.

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

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

```
  \MT@if@list@exists{%
    \ifMT@nofamily
      \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
        \MT@info@nl{Loading generic protrusion settings for font family\MessageBreak
                    `\MT@family' (encoding: \MT@encoding).\MessageBreak
                    For optimal results, create family-specific settings.\MessageBreak
                    See the microtype manual for details}%
        \MT@glet@nc{\MT@encoding-\MT@family-settings}\@empty
      }%
    \fi
    \MT@get@opt
    \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

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

Set an input encoding?

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

Load additional lists?

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

Load the main list.

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

`\MT@set@all@pr`

Set all protrusion codes of the font.

```
\def\MT@set@all@pr#1#2{%
⟨debug⟩\MT@dinfo@nl{3}{-- lp/rp: setting all to #1/#2}%
  \let\MT@temp\@empty
  \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1}}%
  \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2}}%
  \MT@do@font\MT@temp
}
```

`\MT@reset@pr@codes@`
`\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.

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

`\MT@the@pr@code`
`\MT@the@pr@code@tr`

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`.

```
\def\MT@the@pr@code{\@tempcntb}
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex6
⟨luatex-def⟩\MT@requires@luatex3
  {\def\MT@the@pr@code@tr{%
    \numexpr\@tempcntb+\MT@letterspace@/2\relax
  }
}\relax
⟨/pdftex-def|luatex-def⟩
```

`\MT@set@codes`

Split up the values and set the codes.

```
\def\MT@set@codes#1,{%
  \ifx\relax#1\@empty\else
    \MT@split@codes #1==\relax
    \expandafter\MT@set@codes
  \fi
}
```

`\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.

```
\def\MT@split@codes#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
⟨pdftex-def|luatex-def⟩    \ifnum\MT@char > \m@ne
⟨xetex-def⟩    \ifx\MT@char\@empty \else
      \MT@get@char@unit
      \csname MT@\MT@feat @split@val\endcsname#2\relax
    \fi
  \fi
}
```

`\MT@pr@split@val`

```
\def\MT@pr@split@val#1,#2\relax{%
  \def\@tempb{#1}%
  \MT@ifempty\@tempb\relax{%
    \MT@scale@to@em
    \lpcode\MT@font\MT@char=\MT@the@pr@code
⟨debug⟩\MT@dinfo@nl{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char\space: [#1]}%
  }%
  \def\@tempb{#2}%
  \MT@ifempty\@tempb\relax{%
    \MT@scale@to@em
    \rpcode\MT@font\MT@char=\MT@the@pr@code
⟨debug⟩\MT@dinfo@nl{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char\space: [#2]}%
  }%
```

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

```
  \MT@ifdefined@c@T\MT@pr@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c
        {MT@inh@\MT@pr@inh@name @\MT@char @}%
        \MT@set@pr@heirs
    }%
  }%
}
```

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

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

```
⟨pdftex-def⟩\MT@requires@pdftex3{
\def\MT@scale@to@em{%
  \@tempcntb=\MT@count\relax
```

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

```
  \MT@scale\@tempcntb \@tempb \MT@dimen@six
  \ifnum\@tempcntb=\z@ \else
    \MT@scale@factor
  \fi
}
```

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

```
\def\MT@get@charwd{%
⟨*pdftex-def⟩
^^X  \MT@count=\fontcharwd\MT@font\MT@char\relax
^^Q  \setbox\z@=\hbox{\MT@font \char\MT@char}%
^^Q  \MT@count=\wd\z@
⟨/pdftex-def⟩
⟨luatex-def⟩  \MT@count=\fontcharwd\MT@font\MT@char\relax
```

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

```
⟨*xetex-def⟩
  \ifnum\MT@char@<\z@
    \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}%
    \MT@count=\wd\z@
  \else
    \MT@count=\fontcharwd\MT@font\MT@char@\relax
  \fi
⟨/xetex-def⟩
  \ifnum\MT@count=\z@ \MT@info@missing@char \fi
}
```

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.

```
⟨*pdftex-def⟩
\MT@requires@pdftex6{
  \g@addto@macro\MT@get@charwd{%
    \MT@ifdefined@c@T\MT@letterspace@
      {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
  }
}\relax
}{
```

No adjustment with versions 0.14f and 0.14g.

```
\def\MT@scale@to@em{%
  \MT@count=\@tempb\relax
  \ifnum\MT@count=\z@ \else
    \MT@scale@factor
  \fi
}
```

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

```
\def\MT@get@charwd{\MT@count=\MT@dimen@six}
}
⟨/pdftex-def⟩
⟨/pdftex-def|xetex-def|luatex-def⟩
```

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

```
⟨*package⟩
\def\MT@get@font@dimen#1{%
  \ifnum\fontdimen#1\MT@font=\z@
    \MT@warning@nl{Font `\MT@@font' does not specify its\MessageBreak
      \@backslashchar fontdimen #1 (it's zero)!\MessageBreak
      You should use a different `unit' for \MT@curr@list@name}%
  \else
    \MT@count=\fontdimen#1\MT@font
  \fi
}
```

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

```
\def\MT@info@missing@char{%
  \MT@info@nl{Character `\the\MT@toks'
^^X   \ifnum\MT@char@<\z@ is missing\else
^^X     \iffontchar\MT@font\MT@char@
                 has a width of 0pt
^^X      \else is missing\fi\fi
^^Q    \MessageBreak (it's probably missing)
    \MessageBreak in font `\MT@@font'.\MessageBreak
    Ignoring protrusion settings for this character}%
}
```

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

```
\def\MT@scale@factor{%
  \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
    \expandafter\MT@scale\expandafter \@tempcntb
      \csname MT@\MT@feat @factor@\endcsname \@m
  \fi
  \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
    \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
  \else
    \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
      \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
    \fi
  \fi
}
```

`\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.

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

`\MT@get@opt`

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

```
\def\MT@get@opt{%
  \MT@set@listname
```

`\MT@pr@factor@`
`\MT@sp@factor@`
`\MT@kn@factor@`

Apply a factor?

```
  \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
    \MT@let@nn{MT@\MT@feat @factor@}
        {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
    \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
                    \number\csname MT@\MT@feat @factor@\endcsname/1000}%
  }{%
    \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
  }%
```

`\MT@pr@unit@`
`\MT@sp@unit@`
`\MT@kn@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@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
    \MT@let@nn{MT@\MT@feat @unit@}%
        {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
    \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
      \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                      relative to character widths}%
    \else
      \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
        \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                        relative to width of space}%
      \fi
    \fi
  }{%
    \MT@let@nn{MT@\MT@feat @unit@}{MT@\MT@feat @unit}%
  }%
```

`\MT@get@space@unit`
`\MT@get@char@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.

```
  \let\MT@get@char@unit\relax
  \let\MT@get@space@unit\@gobble
  \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
    \let\MT@get@char@unit\MT@get@charwd
  \else
    \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
      \let\MT@get@space@unit\MT@get@font@dimen
    \else
      \MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}%
    \fi
  \fi
```

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

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

`\MT@get@unit`
`\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.

```
\def\MT@get@unit#1{%
  \expandafter\MT@get@unit@#1 e!\@nil
  \ifx\x\@empty\else\let#1\x\fi
  \@defaultunits\@tempdima#1 pt\relax\@nnil
  \ifdim\@tempdima=\z@
    \MT@warning@nl{%
      Cannot set \@nameuse{MT@abbr@\MT@feat} factors relative to zero\MessageBreak
      width. Setting factors of list `\@nameuse{MT@\MT@feat @c@name}'\MessageBreak
      relative to character widths instead}%
    \let#1\@empty
    \let\MT@get@char@unit\MT@get@charwd
  \else
    \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} factors relative
                    to \the\@tempdima}%
    \MT@count=\@tempdima\relax
  \fi
}
\def\MT@get@unit@#1e#2#3\@nil{%
  \ifx\\#3\\\let\x\@empty \else
    \if m#2%
      \edef\x{#1\fontdimen6\MT@font}%
    \else
      \if x#2%
        \edef\x{#1\fontdimen5\MT@font}%
      \fi
    \fi
  \fi
}
```

`\MT@set@inputenc`

The configurations may be under the regime of an input encoding.

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

`\MT@cat`

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

```
  \def\MT@cat{#1}%
```

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

`\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.

```
\MT@addto@setup{%
  \@ifpackageloaded{inputenc}{%
    \@ifpackagelater{inputenc}{2006/02/22}{%
      \def\MT@set@inputenc@{%
        \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
          \MT@load@inputenc
      }%
    }{%
      \let\MT@set@inputenc@\MT@load@inputenc
    }%
  }{%
    \def\MT@set@inputenc@{%
      \MT@warning@nl{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
          \MessageBreak package isn't loaded. Ignoring input encoding}%
    }%
  }%
}
```

`\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.

```
\def\MT@load@inputenc{%
  \MT@cfg@catcodes
⟨debug⟩\MT@dinfo@nl{1}{loading input encoding: \@nameuse{\@tempa}}%
  \inputencoding{\@nameuse{\@tempa}}%
}
⟨/package⟩
```

`\MT@set@pr@heirs`

Set the inheriting characters.

```
⟨*pdftex-def|xetex-def|luatex-def⟩
\def\MT@set@pr@heirs#1{%
  \lpcode\MT@font #1=\lpcode\MT@font\MT@char\relax
  \rpcode\MT@font #1=\rpcode\MT@font\MT@char\relax
⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
⟨debug⟩\MT@dinfo@nl{4}{;;; lp/rp (#1): \number\lpcode\MT@font\MT@char\space/%
⟨debug⟩                                \number\rpcode\MT@font\MT@char\space}%
}
```

`\MT@preset@pr`
`\MT@preset@pr@`

Preset characters. Presetting them relative to their widths is not allowed.

```
\def\MT@preset@pr{%
  \expandafter\expandafter\expandafter\MT@preset@pr@
    \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
}
\def\MT@preset@pr@#1,#2\@nil{%
  \ifx\MT@pr@unit@\@empty
    \MT@warn@preset@towidth{pr}%
    \let\MT@preset@aux\MT@preset@aux@factor
  \else
    \def\MT@preset@aux{\MT@preset@aux@space2}%
  \fi
  \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
  \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
  \MT@set@all@pr\@tempa\@tempb
}
```

`\MT@preset@aux`
`\MT@preset@aux@factor`
`\MT@preset@aux@space`

Auxiliary macro for presetting. Store value ⟨*#1*⟩ in macro ⟨*#2*⟩.

```
\def\MT@preset@aux@factor#1#2{%
  \@tempcntb=#1\relax
  \MT@scale@factor
  \edef#2{\number\@tempcntb}%
}
\def\MT@preset@aux@space#1#2#3{%
  \def\@tempb{#2}%
  \MT@get@space@unit#1%
  \MT@scale@to@em
  \edef#3{\number\@tempcntb}%
}
```

\MT@warn@preset@towidth

```
\def\MT@warn@preset@towidth#1{%
  \MT@warning@nl{%
    Cannot preset characters relative to their widths\MessageBreak
    for \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'. Presetting them%
    \MessageBreak relative to 1em instead}%
}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

14.2.2 Expansion

\MT@expansion Set up for expansion?

```
⟨*pdftex-def|luatex-def⟩
\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).

```
\def\MT@set@ex@codes@s{%
  \MT@if@list@exists{%
    \MT@get@ex@opt
    \let\MT@get@char@unit\relax
    \MT@reset@ef@codes
    \MT@get@inh@list
    \MT@set@inputenc{c}%
    \MT@load@list\MT@ex@c@name
    \MT@set@listname
    \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
    \expandafter\MT@set@codes\@tempc,\relax,%
    \MT@expandfont
  }\relax
}
⟨/pdftex-def|luatex-def⟩
```

\MT@set@ex@codes@n If, on the other hand, all characters should be expanded by the same amount, we only take the first optional argument to `\SetExpansion` into account.

\ifMT@nonselected We need this boolean in `\MT@if@list@exists` so that no warning for missing lists will be issued.

```
⟨package⟩\newif\ifMT@nonselected
```

```
⟨*pdftex-def|luatex-def⟩
\def\MT@set@ex@codes@n{%
  \MT@nonselectedtrue
  \MT@if@list@exists
    \MT@get@ex@opt
  {%
    \let\MT@stretch@  \MT@stretch
    \let\MT@shrink@   \MT@shrink
    \let\MT@step@     \MT@step
    \let\MT@auto@     \MT@auto
    \let\MT@ex@factor@\MT@ex@factor
  }%
  \MT@reset@ef@codes
  \MT@expandfont
  \MT@nonselectedfalse
}
```

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

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

`\MT@expandfont`

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

```
⟨*luatex-def⟩
\MT@requires@luatex3{
\MT@requires@luatex4{\let\pdffontexpand\expandglyphsinfont}\relax
\ifnum\luatexversion<79
\def\MT@expandfont{%
  \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ autoexpand\relax
}
\else
\def\MT@expandfont{%
  \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@\relax
}
\fi
}{
⟨/luatex-def⟩
\def\MT@expandfont{%
  \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
}
⟨luatex-def⟩}
```

`\MT@set@all@ex`
`\MT@reset@ef@codes@`

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

```
\def\MT@set@all@ex#1{%
⟨debug⟩\MT@dinfo@nl{3}{-- ex: setting all to \number#1}%
  \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
}
\def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}
```

`\MT@reset@ef@codes`

However, this is only necessary for pdfTeX versions prior to 1.20, or LuaTeX < 0.90 (actually, I think, 0.87).

```
⟨pdftex-def⟩\MT@requires@pdftex4
⟨luatex-def⟩\MT@requires@luatex5
{
  \def\MT@reset@ef@codes{%
    \ifnum\MT@ex@factor@=\@m \else
      \MT@reset@ef@codes@
    \fi
  }
}{
  \let\MT@reset@ef@codes\MT@reset@ef@codes@
}
```

`\MT@ex@split@val`

There's only one number per character.

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

Take an optional factor into account.

```
  \ifnum\MT@ex@factor@=\@m \else
    \MT@scale\@tempcntb \MT@ex@factor@ \@m
  \fi
  \ifnum\@tempcntb > \MT@ex@max
    \MT@warn@ex@too@large\MT@ex@max
  \else
    \ifnum\@tempcntb < \MT@ex@min
      \MT@warn@ex@too@large\MT@ex@min
    \fi
  \fi
  \efcode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%
```

Heirs, heirs, I love thy heirs.

```
  \MT@ifdefined@c@T\MT@ex@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@ex@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
    }%
  }%
}
```

`\MT@warn@ex@too@large`

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

`\MT@get@ex@opt`
`\MT@ex@factor@`
`\MT@stretch@`
`\MT@shrink@`
`\MT@step@`
`\MT@auto@`

Apply different values to this font?

```
\def\MT@get@ex@opt{%
  \MT@set@listname
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @factor}{%
    \MT@let@cn\MT@ex@factor@{MT@ex@c@\MT@ex@c@name @factor}%
    \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
  }{%
    \let\MT@ex@factor@\MT@ex@factor
  }%
  \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
  \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
  \MT@get@ex@opt@{step}   {Setting expansion step to \number\MT@step@}%
⟨luatex-def⟩  \MT@requires@luatex3\relax{%
  \MT@get@ex@opt@{auto}{\MT@ifstreq{\MT@auto@}{autoexpand}{En}{Dis}abling automatic expansion}%
⟨luatex-def⟩  }%
  \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @preset}{%
    \MT@preset@ex
    \let\MT@reset@ef@codes\relax
  }%
}
```

`\MT@get@ex@opt@`

```
\def\MT@get@ex@opt@#1#2{%
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @#1}{%
    \MT@let@nn{MT@#1@}{MT@ex@c@\MT@ex@c@name @#1}%
    \MT@vinfo{... : #2}%
  }{%
    \MT@let@nn{MT@#1@}{MT@#1}%
  }%
}
```

`\MT@set@ex@heirs`

```
\def\MT@set@ex@heirs#1{%
  \efcode\MT@font#1=\efcode\MT@font\MT@char
⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
⟨debug⟩\MT@dinfo@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
}
```

`\MT@preset@ex`

```
\def\MT@preset@ex{%
  \@tempcntb=\csname MT@ex@c@\MT@ex@c@name @preset\endcsname\relax
  \MT@scale@factor
  \MT@set@all@ex\@tempcntb
}
⟨/pdftex-def|luatex-def⟩
```

14.2.3 Interword spacing (glue)

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

```
⟨*pdftex-def⟩
\MT@requires@pdftex6{
\def\MT@spacing{\MT@maybe@do{sp}}
```

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

```
\def\MT@set@sp@codes{%
  \MT@if@list@exists{%
    \MT@get@opt
    \MT@reset@sp@codes
    \MT@get@inh@list
    \MT@set@inputenc{c}%
    \MT@load@list\MT@sp@c@name
    \MT@set@listname
    \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
    \expandafter\MT@set@codes\@tempc,\relax,%
  }\MT@reset@sp@codes
}
```

`\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).

```
\def\MT@sp@split@val#1,#2,#3\relax{%
  \def\@tempb{#1}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit2%
    \MT@scale@to@em
    \knbscode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
  }%
  \def\@tempb{#2}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit3%
    \MT@scale@to@em
    \stbscode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
  }%
  \def\@tempb{#3}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit4%
    \MT@scale@to@em
    \shbscode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
  }%
  \MT@ifdefined@c@T\MT@sp@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@sp@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@sp@inh@name @\MT@char @}\MT@set@sp@heirs
    }%
  }%
}
```

`\MT@set@sp@heirs`

```
\def\MT@set@sp@heirs#1{%
  \knbscode\MT@font#1=\knbscode\MT@font\MT@char
  \stbscode\MT@font#1=\stbscode\MT@font\MT@char
  \shbscode\MT@font#1=\shbscode\MT@font\MT@char
⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
⟨debug⟩\MT@dinfo@nl{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
⟨debug⟩        \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
}
```

`\MT@set@all@sp`
`\MT@reset@sp@codes`
`\MT@reset@sp@codes@`

```
\def\MT@set@all@sp#1#2#3{%
⟨debug⟩\MT@dinfo@nl{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
  \let\MT@temp\@empty
  \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
  \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
  \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
  \MT@do@font\MT@temp
}
\def\MT@reset@sp@codes@{\MT@set@all@sp\z@\z@\z@}
\let\MT@reset@sp@codes\relax
```

`\MT@preset@sp`
`\MT@preset@sp@`

```
\def\MT@preset@sp{%
  \expandafter\expandafter\expandafter\MT@preset@sp@
    \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
}
\def\MT@preset@sp@#1,#2,#3\@nil{%
  \ifx\MT@sp@unit@\@empty
    \MT@warn@preset@towidth{sp}%
    \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
    \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
    \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
  \else
    \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
    \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
    \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
  \fi
  \MT@set@all@sp\@tempa\@tempc\@tempb
}
}\relax
```

14.2.4 Additional kerning

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

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

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

```
\def\MT@set@kn@codes{%
  \MT@if@list@exists{%
    \MT@get@opt
    \MT@reset@kn@codes
    \MT@get@inh@list
    \MT@set@inputenc{c}%
    \MT@load@list\MT@kn@c@name
    \MT@set@listname
    \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
    \expandafter\MT@set@codes\@tempc,\relax,%
  }\MT@reset@kn@codes
}
```

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

```
\def\MT@kn@split@val#1,#2\relax{%
  \def\@tempb{#1}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit2%
    \MT@scale@to@em
    \knbccode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
  }%
  \def\@tempb{#2}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit2%
    \MT@scale@to@em
    \knaccode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
  }%
  \MT@ifdefined@c@T\MT@kn@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{%
      \MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
    }%
  }%
}
```

`\MT@set@kn@heirs`

```
\def\MT@set@kn@heirs#1{%
  \knbccode\MT@font#1=\knbccode\MT@font\MT@char
  \knaccode\MT@font#1=\knaccode\MT@font\MT@char
⟨debug⟩\MT@dinfo@nl{2}{-- heir of \MT@char: #1}%
⟨debug⟩\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
⟨debug⟩                               \number\knaccode\MT@font\MT@char}%
}
```

`\MT@set@all@kn`
`\MT@reset@kn@codes`
`\MT@reset@kn@codes@`

```
\def\MT@set@all@kn#1#2{%
⟨debug⟩\MT@dinfo@nl{3}{-- knac/knbc: setting all to #1/#2}%
  \let\MT@temp\@empty
  \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbccode\MT@font\@tempcnta=#1\relax}}%
  \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knaccode\MT@font\@tempcnta=#2\relax}}%
  \MT@do@font\MT@temp
}
\def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
\let\MT@reset@kn@codes\relax
```

`\MT@preset@kn`
`\MT@preset@kn@`

```
\def\MT@preset@kn{%
  \expandafter\expandafter\expandafter\MT@preset@kn@
    \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
}
\def\MT@preset@kn@#1,#2\@nil{%
  \ifx\MT@kn@unit@\@empty
    \MT@warn@preset@towidth{kn}%
    \let\MT@preset@aux\MT@preset@aux@factor
  \else
    \def\MT@preset@aux{\MT@preset@aux@space2}%
  \fi
  \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
  \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
  \MT@set@all@kn\@tempa\@tempb
}
}\relax
⟨/pdftex-def⟩
```

14.2.5 Tracking

This only works with pdfTEX 1.40 or LuaTEX 0.62.

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex6
⟨luatex-def⟩\MT@requires@luatex3
{
```

`\MT@tracking`
`\MT@tracking@`
`\MT@tr@font@list`

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

```
\let\MT@tr@font@list\@empty
\def\MT@tracking@{%
  \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
  \ifMT@inlist@\else
    \MT@maybe@do{tr}%
    \ifMT@do\else
      \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
    \fi
  \fi
}
⟨/pdftex-def|luatex-def⟩
⟨pdftex-def|luatex-def|letterspace⟩\let\MT@tracking
⟨pdftex-def|luatex-def⟩  \MT@tracking@
⟨letterspace⟩  \relax
```

`\MT@set@tr@codes`

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

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

```
⟨*pdftex-def|luatex-def|letterspace⟩
\def\MT@set@tr@codes{%
⟨*pdftex-def|luatex-def⟩
  \MT@vinfo{Tracking font `\MT@@font'\on@line}%
  \MT@ifdefined@n@TF{\MT@@font-fake6}{%
    \expandafter\ifx\csname\MT@@font-fake6\endcsname\@empty
      \MT@warning@nl{%
        Font `\MT@@font' does not specify its\MessageBreak
        \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
        \@nameuse{MT@abbr@\MT@feat} will not work with this font}%
      \MT@glet@nc{\MT@@font-fake6}\relax
    \fi
  }{%
  \MT@if@list@exists
    \MT@get@tr@opt
    \relax
⟨/pdftex-def|luatex-def⟩
  \MT@ifdefined@c@TF\MT@letterspace@\relax{\let\MT@letterspace@\MT@letterspace}%
  \ifnum\MT@letterspace@=\z@
```

Zero tracking requires special treatment.

```
    \MT@set@tr@zero
  \else
⟨pdftex-def|luatex-def⟩    \MT@vinfo{... Tracking by \number\MT@letterspace@}%
```

Letterspacing only works in PDF mode.

```
    \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.

```
    \xdef\MT@lsfont{\csname\expandafter\string\font@name
                           /\number\MT@letterspace@ ls\endcsname}%
    \expandafter\ifx\MT@lsfont\relax
⟨debug⟩\MT@dinfo@nl{1}{... new letterspacing instance}%
```

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

```
      \MT@get@ls@basefont
```

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

```
⟨*luatex-def|letterspace⟩
      \MT@if@luaotf@font{%
⟨luatex-def&debug⟩\MT@dinfo@nl{1}{... luaotf font: \MessageBreak
⟨luatex-def&debug⟩         \expandafter\fontname\font@name}%
        \ifnum\MT@letterspace@<\z@\def\MT@minus{-}\else\let\MT@minus\@empty\fi
        \global\expandafter\font\MT@lsfont=%
          \expandafter\MT@exp@two@c\expandafter\MT@ls@fontspec@font
            \expandafter\fontname\expandafter\font@name\space \@nil
      }{%
⟨/luatex-def|letterspace⟩
⟨luatex-def&debug⟩\MT@dinfo@nl{1}{... legacy font}%
      \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
⟨luatex-def|letterspace⟩      }%
```

Scale interword spacing (not configurable in `letterspace`).

```
⟨*pdftex-def|luatex-def⟩
      \MT@ifdefined@c@TF\MT@tr@ispace
        {\let\@tempa\MT@tr@ispace}%
        {\edef\@tempa{\MT@letterspace@*,,}}%
      \MT@ifdefined@c@TF\MT@tr@ospace
        {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
        {\edef\@tempa{\@tempa,,,}}%
      \expandafter\MT@tr@set@space\@tempa,%
⟨/pdftex-def|luatex-def⟩
⟨*letterspace⟩
      % spacing = {<letterspace amount>*,,}
      \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
                                            * \fontdimen2\MT@lsfont/1000\relax
⟨/letterspace⟩
```

Adjust outer kerning (`microtype` only).

```
⟨*pdftex-def|luatex-def⟩
      \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,*}}%
      \expandafter\MT@tr@set@okern\@tempa,%
```

Disable ligatures (not configurable in `letterspace`).

```
      \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
⟨/pdftex-def|luatex-def⟩
⟨*letterspace⟩
      % no ligatures = {f}
      \tagcode\MT@lsfont`f=\m@ne
⟨/letterspace⟩
```

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

```
⟨luatex-def|letterspace⟩      \MT@if@luaotf@font\relax{%
⟨debug⟩\MT@dinfo@nl{2}{... compensating for tracking (\number\MT@letterspace@)}%
      \MT@do@font{\lpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax
                  \rpcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%
      \let\MT@the@pr@code\MT@the@pr@code@tr
⟨luatex-def|letterspace⟩     }%
    \fi
```

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

```
    \aftergroup\MT@set@lsfont
⟨pdftex-def|luatex-def⟩    \let\MT@font\MT@lsfont
⟨luatex-def⟩    \MT@if@luaotf@font\MT@font\relax
```

`\MT@set@curr@ls`
`\MT@curr@ls`

We need to remember the current letterspacing amount (for `\lslig`).

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

```
⟨*pdftex-def|luatex-def⟩
    \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax
    \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%
    \MT@tr@outer@l
⟨/pdftex-def|luatex-def⟩
```

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

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

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

```
⟨*pdftex-def|luatex-def⟩
    \else
      \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
                      \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
      \ifdim\MT@outer@kern=\z@\else \MT@ls@outer@k \fi
      \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
                      \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
⟨/pdftex-def|luatex-def⟩
⟨*letterspace⟩
      \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
      \MT@afteraftergroup{%
        \MT@set@curr@ok
        \noexpand\MT@ls@outer@k
      }%
⟨/letterspace⟩
    \fi
⟨*pdftex-def|luatex-def⟩
```

`\MT@set@curr@ok`

Carry the outer kerning amount to outside the next group, then set outer spacing (which will set kerning, if no space follows).

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

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

```
    \MT@afteraftergroup{%
      \MT@set@curr@os
      \MT@set@curr@ok
      \noexpand\MT@tr@outer@r
    }%
⟨/pdftex-def|luatex-def⟩
  \fi
⟨pdftex-def|luatex-def⟩ }%
}
```

`\MT@afteraftergroup`

This helper macro carries stuff outside of the current group to the end of the next group, but will then respect grouping, which is crucial for nested letterspacing. (Following an idea of Will Robertson.)

```
\def\MT@afteraftergroup#1{%
⟨!letterspace⟩ \MT@maybe@gobble@with@tikz{%
    \MT@ifdefined@n@TF{MT@aftergroup@\number\currentgrouplevel}\relax{%
      \MT@exp@cs\xdef{MT@aftergroup@\number\currentgrouplevel}%
        {\MT@exp@cs\MT@glet{MT@aftergroup@\number\currentgrouplevel}\noexpand\@undefined#1}%
      \expandafter\aftergroup\expandafter\aftergroup\MT@exp@cs\aftergroup
        {MT@aftergroup@\number\currentgrouplevel}%
    }%
⟨!letterspace⟩ }%
}
⟨/pdftex-def|luatex-def|letterspace⟩
```

`\MT@ls@fontspec@colon`
`\MT@ls@fontspec@font`

Add the `kernfactor` feature to a font loaded by `fontspec` (we might have to add the colon ourselves).

```
⟨*luatex-def|letterspace⟩
\def\MT@ls@fontspec@colon#1:#2:#3:#4\@nil{\ifx\\#3\\#1:#2\else#1:#2:#3\fi}
\def\MT@ls@fontspec@font#1 #2\@nil{%
  "\MT@ls@fontspec@colon#1:::\relax\@nil
    kernfactor=\MT@minus \ifnum\MT@letterspace@=1000 1\else 0.%
         \ifnum\MT@minus\MT@letterspace@<100 0\fi
         \ifnum\MT@minus\MT@letterspace@<10  0\fi
        \number\MT@minus\MT@letterspace@ \fi;"
  \ifx\\#2\\ at \f@size pt\else#2\fi\relax
}
⟨/luatex-def|letterspace⟩
```

`\MT@get@tr@opt`

Various settings (only for the `microtype` version).

```
⟨*pdftex-def|luatex-def⟩
\def\MT@get@tr@opt{%
  \MT@set@listname
  \let\MT@tr@factor@\@m
```

`\MT@tr@unit@`

Different unit (for `letterspace` and/or (`outer`)`spacing`)?

```
  \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @unit}{%
    \MT@let@cn\MT@tr@unit@{MT@tr@c@\MT@tr@c@name @unit}%
    \ifdim\MT@tr@unit@=1em
      \let\MT@tr@unit@\@undefined
    \else
      \MT@get@unit\MT@tr@unit@
    \fi
  }%
  \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name}{%
    \MT@let@cn\MT@letterspace{MT@tr@c@\MT@tr@c@name}%
    \MT@ifdefined@c@T\MT@tr@unit@{%
      \let\@tempb\MT@letterspace
      \MT@scale@to@em
      \edef\MT@letterspace{\number\@tempcntb}%
    }%
  }%
```

`\MT@tr@ispace`

Adjust interword spacing.

`\MT@tr@ospace`

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

`\MT@tr@okern`

Adjust outer kerning.

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

`\MT@tr@ligatures`

Which ligatures should we disable (empty means all, undefined none)?

```
  \MT@get@tr@opt@{noligatures} {ligatures}%
}
```

`\MT@get@tr@opt@`

```
\def\MT@get@tr@opt@#1#2{%
  \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @#1}%
    {\MT@let@nn{MT@tr@#2}{MT@tr@c@\MT@tr@c@name @#1}}%
}
⟨/pdftex-def|luatex-def⟩
```

`\MT@set@lsfont`

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

```
⟨*pdftex-def|luatex-def|letterspace⟩
⟨plain⟩\MT@requires@latex2{
\def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}
```

`\lsstyle`

Disable the tests whether the font should be letterspaced, then trigger the setup. Only `\textls` can be used in math mode (`\lsstyle` may be used inside another text switch, of course). Still, we have to ensure that math fonts are set up again.

Setting `\glb@currsize` globally to `\@empty` (our previous solution) could throw us into an infinite loop (e.g., with the `psnfss` packages, via `\every@math@size`), so we issue `\glb@settings` instead. However, in certain situations, we may still miss some math fonts, so let's try to also enforce it by emptying `\glb@currsize`, fingers crossed. The overhead seems small.

```
\DeclareRobustCommand\lsstyle{%
  \not@math@alphabet\lsstyle\textls
  \let\glb@currsize\@empty
⟨pdftex-def|luatex-def⟩  \MT@maybe@gobble@with@tikz{\aftergroup\glb@settings}%
⟨pdftex-def|luatex-def⟩  \def\MT@feat{tr}%
  \let\MT@tracking\MT@set@tr@codes
  \selectfont
}
```

Now the definitions for the `letterspace` package with plain TeX.

```
⟨*plain⟩
}{
\def\MT@set@lsfont{\MT@lsfont}
\def\lsstyle{%
  \begingroup
  \escapechar\m@ne
  \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
  \MT@set@tr@codes
  \endgroup
}
\let\textls\@undefined
\let\lslig\@undefined
}
⟨/plain⟩
```

`\lslig`

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

```
\DeclareRobustCommand\lslig[1]{%
  {\MT@ifdefined@c@TF\MT@curr@ls{%
     \escapechar\m@ne
     \MT@get@ls@basefont
     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
     \kern\MT@outer@kern
     \font@name #1%
     \kern\MT@outer@kern
  }{#1}}%
}
```

`\MT@ls@basefont`
`\MT@get@ls@basefont`

pdfTeX cannot letterspace fonts that already are letterspaced. Therefore, we have to save the base font in `\`⟨*font name*⟩`@base`.

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

```
\def\MT@get@ls@basefont{%
  \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
  \expandafter\ifx\MT@ls@basefont\relax
    \MT@exp@two@c\MT@glet\MT@ls@basefont\font@name
  \else
⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
    \MT@exp@two@c\let\font@name\MT@ls@basefont
  \fi
}
```

`\MT@set@lsbasefont`
`\MT@set@tr@zero`

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.

```
\def\MT@set@lsbasefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
\def\MT@set@tr@zero{%
⟨debug⟩\MT@dinfo@nl{1}{... zero tracking}%
  \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
  \expandafter\ifx\MT@ls@basefont\relax \else
⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
    \aftergroup\MT@set@lsbasefont
  \fi
}
⟨/pdftex-def|luatex-def|letterspace⟩
```

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

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex7{
  \def\MT@tr@noligatures{%
    \ifx\MT@tr@ligatures\@empty
      \MT@noligatures@\MT@lsfont\@undefined
    \else
      \MT@noligatures@\MT@lsfont\MT@tr@ligatures
    \fi
  }
⟨*pdftex-def⟩
}{
  \def\MT@tr@noligatures{%
    \MT@warning@nl{%
      Disabling selected ligatures is only possible since\MessageBreak
      pdftex 1.40.4. Disabling all ligatures instead}%
    \MT@glet\MT@tr@noligatures\relax
  }
}
⟨/pdftex-def⟩
```

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

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

`\MT@tr@set@space` Adjust interword spacing (`\fontdimen` 2,3,4) for inner and outer space. For inner spacing, the font dimensions will be adjusted, the settings for outer spacing will be remembered in a macro.

```
\def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
⟨debug⟩\MT@dinfo@nl2{... orig. space: \the\fontdimen2\MT@lsfont,
⟨debug⟩    \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
⟨debug⟩    \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
  \let\MT@temp\@empty
  \MT@tr@set@space@{#1}{#4}{2}\@empty
  \MT@tr@set@space@{#2}{#5}{3}\@plus
  \MT@tr@set@space@{#3}{#6}{4}\@minus
  \MT@glet@nc{MT@outer@space\expandafter\string\font@name}\MT@temp
⟨debug⟩\MT@dinfo@nl2{... inner space: \the\fontdimen2\MT@lsfont,
⟨debug⟩    \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
⟨debug⟩\MT@dinfo@nl2{... outer space: \MT@temp}%
}
```

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

```
\def\MT@tr@set@space@#1#2#3#4{%
  \MT@ifempty{#2}{%
    \MT@ifempty{#1}\relax{%
      \MT@tr@set@space@@{#1}{#3}{1000}%
      \fontdimen#3\MT@lsfont=\@tempdima
    }%
    \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
  }{%
    \MT@tr@set@space@@{#2}{#3}{2000}%
    \edef\MT@temp{\MT@temp#4\the\@tempdima}%
    \MT@ifempty{#1}\relax{%
      \MT@tr@set@space@@{#1}{#3}{1000}%
      \fontdimen#3\MT@lsfont=\@tempdima
    }%
  }%
}
```

`\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.

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

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

```
    \ifnum#2=\tw@
      \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
    \fi
  }{%
    \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
    \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
  }%
⟨debug⟩\MT@dinfo@nl3{... : font dimen #2 (#1): \the\@tempdima}%
}
```

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

```
\def\MT@tr@outer@l{%
  \ifhmode
    \ifdim\lastskip>5sp
      \edef\x{\the\lastskip minus 0pt}%
      \setbox\z@\hbox{\MT@outer@space=\x}%
      \ifdim\wd\z@>\z@
⟨debug⟩\MT@dinfo2{[[[ adjusting pre space: \the\MT@outer@space}%
        \unskip \hskip\MT@outer@space\relax
```

Disable left outer kerning.

```
        \let\MT@ls@outer@k\relax
      \else
```

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

```
        \ifdim\lastskip=%
            \ifnum\spacefactor<2000
              \spaceskip
            \else
              \ifdim\xspaceskip=\z@
                \dimexpr\spaceskip+\fontdimen7\font@name\relax
              \else
                \xspaceskip
              \fi
            \fi
⟨debug⟩\MT@dinfo2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
          \unskip \hskip\MT@outer@space\relax
          \let\MT@ls@outer@k\relax
        \fi
      \fi
    \fi
  \fi
}
```

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

```
\def\MT@tr@outer@r{%
  \futurelet\MT@tr@outer@next\MT@tr@outer@r@
}
```

`\MT@if@outer@next`

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

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

`\MT@tr@outer@r@`

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

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

```
  \ifmmode \else
```

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

```
    \ifnum\currentgrouptype=10 \else
      \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
⟨debug⟩\MT@dinfo2{]]] adjusting post space (1): \the\MT@outer@space}%
        \fi}%
      \expandafter\ifcat\expandafter\noexpand\csname MT@tr@outer@next\endcsname\egroup

        \ifhmode\unkern\fi\egroup
        \MT@set@curr@ok \MT@set@curr@os
        \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
      \else
```

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

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

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

```
          \MT@if@outer@next\check@icr{%
            \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}%
          }{%
            \MT@if@outer@next\@sptoken{%
              \def\MT@temp* {\ifhmode\hskip\MT@outer@space
⟨debug⟩\MT@dinfo2{]]] adjusting post space (2): \the\MT@outer@space}%
                \fi}%
            }{%
              \MT@if@outer@next~{%
                \def\MT@temp*~{\nobreak\hskip\MT@outer@space
⟨debug⟩\MT@dinfo2{]]] adjusting post space (3): \the\MT@outer@space}%
                  }%
              }{%
                \MT@if@outer@next\ \relax{%
                  \MT@if@outer@next\space\relax{%
                    \MT@if@outer@next\@xobeysp\relax{%
```

`xspace` requires special treatment.

```
                      \MT@if@outer@next\xspace{%
```

```
                        \def\MT@temp*\xspace{\MT@xspace}%
                      }{%
```

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

```
                        \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k
⟨debug⟩\MT@dinfo2{--- adjusting post kern: \the\MT@outer@kern}%
                          \fi}%
                        \MT@let@nc{MT@tr@outer@next}\relax
      }}}}}}}}\fi
  \fi\fi
  \MT@temp*%
}
```

\MT@tr@outer@icr
\MT@tr@outer@icr@

Helper macros for the italic correction mess.

```
\def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
\def\MT@tr@outer@icr@{%
  \let\@let@token= \MT@tr@outer@next
  \maybe@ic@
}
```

\MT@xspace
\MT@xspace@

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

```
\def\MT@xspace{\futurelet\@let@token\MT@xspace@}
\def\MT@xspace@{\@xspace@firsttrue\@xspace
  \ifdim\lastskip>5sp
    \unskip \hskip\MT@outer@space
  \else
    \ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k \fi
  \fi
}
```

For older pdfTEX versions and LuaTEX, throw an error.

```
}{
  \DeclareRobustCommand\lsstyle{%
    \MT@error{Letterspacing only works with \MT@engine tex version
⟨pdftex-def⟩      1.40%
⟨luatex-def⟩      0.62%
      \MessageBreak or newer}
      {Upgrade \MT@engine tex, or try the `soul' package instead.}%
    \MT@glet\lsstyle\relax
  }
}
```

And for XƎTEX, too.

```
⟨/pdftex-def|luatex-def⟩
⟨*xetex-def⟩
\DeclareRobustCommand\lsstyle{%
  \MT@error{Letterspacing currently doesn't work with xetex}
           {Run pdftex or luatex, or use the `soul' package instead.}%
  \MT@glet\lsstyle\relax
}
⟨/xetex-def⟩
```

\textls
\MT@ls@adjust@

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.

```
⟨*package|letterspace⟩
\DeclareRobustCommand\textls{%
  \@ifstar{\let\MT@ls@adjust@\MT@ls@adjust@empty\MT@textls}%
          {\let\MT@ls@adjust@\MT@ls@adjust@relax\MT@textls}%
}
```

\MT@textls
\MT@letterspace@

This is now almost LATEX’s `\DeclareTextFontCommand`, with the difference that we adjust the outer spacing and kerning also for `\lsstyle`, while LaTeX's text *switches* don't bother about italic correction.

```
\newcommand\MT@textls[2][]{%
  \ifmmode
    \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
  \else
    \hmode@bgroup
      \MT@ls@set@ls{#1}%
      \lsstyle #2%
      \expandafter
    \egroup
  \fi
}
```

`\MT@ls@adjust`
`\MT@ls@adjust@empty`
`\MT@ls@adjust@relax`
`\MT@ls@set@ls`

Set current letterspacing amount and outer kerning. This has to be done inside the same group as the letterspacing command.

```
\def\MT@ls@adjust@empty{\let\MT@ls@adjust\@empty}
\def\MT@ls@adjust@relax{\let\MT@ls@adjust\relax}
\def\MT@ls@set@ls#1{%
  \MT@ifempty{#1}%
    {\let\MT@letterspace@\@undefined}%
    {\KV@@sp@def\MT@letterspace@{#1}%
     \edef\MT@letterspace@{\number\MT@letterspace@}%
     \MT@ls@too@large\MT@letterspace@}%
  \MT@ls@adjust@
}
```

`\MT@ls@too@large`

Test whether letterspacing amount is too large.

```
\def\MT@ls@too@large#1{%
  \ifnum#1>\MT@tr@max
    \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
    \let#1\MT@tr@max
  \else
    \ifnum#1<\MT@tr@min
      \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
      \let#1\MT@tr@min
    \fi
  \fi
}
```

`\MT@outer@kern`
`\MT@tr@set@okern`

This dimen is used for the starred version of `\textls`, for `\lslig` and for adjusted outer kerning.

```
\newdimen\MT@outer@kern
⟨/package|letterspace⟩
⟨*pdftex-def|luatex-def⟩
\def\MT@tr@set@okern#1,#2,{%
  \let\MT@temp\@empty
  \MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
  \MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
  \MT@glet@nc{MT@outer@kern\expandafter\string\font@name}\MT@temp
⟨debug⟩\MT@dinfo@nl2{... outer kerning: (#1,#2)
⟨debug⟩               = \@nameuse{MT@outer@kern\expandafter\string\font@name}}%
}
```

`\MT@tr@set@okern@`

```
\def\MT@tr@set@okern@#1{%
  \MT@test@ast#1*\@nil{%
    \MT@ifdefined@c@TF\MT@tr@unit@
      {\edef\@tempb{#1}\MT@scale@to@em}
      {\@tempcntb=#1\relax}%
    \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
  }{%
    \MT@ifempty\@tempa{\let\@tempa\@m}\relax
    \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
```

```
                      * \fontdimen6\MT@lsfont/2000\relax
  }%
  \advance\@tempdima -\dimexpr \MT@letterspace@ sp
                             * \fontdimen6\MT@lsfont/2000\relax
  \edef\MT@temp{\MT@temp{\the\@tempdima}}%
}
⟨/pdftex-def|luatex-def⟩
```

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

```
⟨*pdftex-def|luatex-def|letterspace⟩
\def\MT@ls@outer@k{%
  \ifhmode
    \ifdim\lastkern=-3sp \unkern
      \ifdim\lastkern=3sp \kern-3sp
        \expandafter\expandafter\expandafter\@gobble
      \else \unkern
        \expandafter\expandafter\expandafter\@firstofone
      \fi
    \else
      \expandafter\@firstofone
    \fi
    {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
  \fi
}
⟨/pdftex-def|luatex-def|letterspace⟩
```

14.2.6 Disabling ligatures

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

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex5{
\def\MT@noligatures{%
  \MT@dotrue
  \let\@tempa\MT@nl@setname
  \MT@map@clist@n{font,encoding,family,series,shape,size}{%
    \MT@ifdefined@n@TF{MT@checklist@##1}%
      {\csname MT@checklist@##1\endcsname}%
      {\MT@checklist@{##1}}%
    {nl}%
  }%
  \ifMT@do
    \MT@noligatures@\MT@font\MT@nl@ligatures
  \fi
}
```

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

```
⟨luatex-def⟩\MT@requires@luatex4{\let\pdfnoligatures\ignoreligaturesinfont}\relax
\def\MT@noligatures@#1#2{%
  \MT@ifdefined@c@TF#2{%
```

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

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

No '`inputenc`' key.

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

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

```
⟨luatex-def⟩          \MT@if@luaotf@font
⟨luatex-def⟩             {\MT@lua{microtype.noligatures([[#1]],[[\MT@char]])}}\relax
        \fi
      }%
      \MT@vinfo{... Disabling ligatures for characters: #2}%
    }{%
      \pdfnoligatures#1%
      \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
          know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
          the font instead}%
    }%
  }{%
    \pdfnoligatures#1%
⟨luatex-def⟩    \MT@if@luaotf@font
⟨luatex-def⟩        {\MT@lua{microtype.noligatures([[#1]],"_all_")}}\relax
    \MT@vinfo{... Disabling all ligatures}%
  }%
}
⟨pdftex-def⟩}\relax
⟨/pdftex-def|luatex-def⟩
```

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

```
⟨*luafile⟩
microtype.ligs = microtype.ligs or { }

local function noligatures(fontcs,liga)
  local fontcs = match(fontcs,"([^ ]+)")
  microtype.ligs[fontcs] = microtype.ligs[fontcs] or { }
  table.insert(microtype.ligs[fontcs],liga)
end
microtype.noligatures = noligatures

local function keepligature(c)
  local nodedirect = node.direct
  local getfield   = nodedirect.getfield
  local getfont    = nodedirect.getfont
  local f,ch
  if type(c) == "userdata" then -- in older luaotfload versions, c was a node
    f  = c.font
    ch = c.components.char
  else                          -- since 2.6, c is a (direct node) number
    f  = getfont(c)
    ch = getfield(getfield(c,"components"),"char")
  end
--  if ch then -- should always be true
  local ligs = microtype.ligs[match(tex.fontidentifier(f),"\\([^ ]+)")]
  if ligs then
    for _,lig in pairs(ligs) do
      if lig == "_all_" or tonumber(lig) == ch then
        return false
      end
    end
  end
  return true
--  end
end

if luaotfload and luaotfload.letterspace then
  if luaotfload.letterspace.keepligature then
```

```
    microtype.info("overwriting function `keepligature'")
  end
  luaotfload.letterspace.keepligature = keepligature
end

⟨/luafile⟩
```

14.2.7 Loading the configuration

\MT@load@list

Recurse through the lists to be loaded.

```
⟨*package⟩
\def\MT@load@list#1{%
  \edef\@tempa{#1}%
  \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
  \MT@ifstreq\@tempa\@tempb{%
    \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempa' cannot load itself}{}%
  }{%
    \ifx\@tempb\relax \else
      \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
        \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `\@tempb'}%
        \begingroup
          \MT@load@list\@tempb
        \endgroup
        \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
          \noexpand\MessageBreak`\@tempb'}%
        \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
        \expandafter\MT@set@codes\@tempc,\relax,%
      }{%
        \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempb' undefined.\MessageBreak
                    Cannot load it from list `\@tempa'}{}%
      }%
    \fi
  }%
}
```

\MT@find@file
\MT@file@list

Micro-typographic settings may be written into a file `mt-`⟨*font family*⟩`.cfg`. We must also record whether we've already loaded the file.

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

Check for existence of the file only once.

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

```
    \MT@begin@catcodes
      \let\MT@begin@catcodes\relax
      \let\MT@end@catcodes\relax
      \InputIfFileExists{mt-#1.cfg}{%
        \edef\MT@curr@file{mt-#1.cfg}%
        \MT@vinfo{... Loading configuration file \MT@curr@file}%
        \MT@xadd\MT@file@list{#1,}%
      }{%
        \MT@get@basefamily#1\@empty\@empty\@empty\@nil
        \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
        \ifMT@inlist@
          \MT@xadd\MT@file@list{#1,}%
        \else
          \InputIfFileExists{mt-\@tempa.cfg}{%
            \edef\MT@curr@file{mt-\@tempa.cfg}%
            \MT@vinfo{... Loading configuration file \MT@curr@file}%
            \MT@xadd\MT@file@list{\@tempa,#1,}%
          }{%
            \MT@vinfo{... No configuration file mt-#1.cfg}%
            \MT@xadd\MT@file@list{#1,}%
          }%
        \fi
      }%
    \endgroup
  \fi
}
```

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

```
\def\MT@cfg@catcodes{%
  \makeatletter
  \catcode`\^7%
  \catcode`\ 9%
  \catcode`\^^I9%
  \catcode`\^^M9%
  \catcode`\\\z@
  \catcode`\{\@ne
  \catcode`\}\tw@
  \catcode`\#6%
  \catcode`\%14%
  \MT@map@tlist@n
    {\!\"\$\&\'\(\)\*\+\,\-\.\/\:\;\<\=\>\?\[\]\_\`\|\~}%
    \@makeother
}
```

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

```
\def\MT@begin@catcodes{%
  \begingroup
  \MT@cfg@catcodes
}
```

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

```
\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.

```
\def\MT@get@basefamily#1#2#3#4\@nil{%
  \ifx\@empty#4%
    \def\@tempa{#1#2#3}%
  \else
    \let\@tempa\@empty
    \edef\@tempb{#1#2#3#4}%
    \expandafter\MT@get@basefamily@\@tempb\@nil
  \fi
}
```

`\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`'.

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	●	-	●	-	●	-	●	-	●	-	●	-	●	-	●	-

```
\def\MT@get@basefamily@#1#2\@nil{%
  \edef\@tempa{\@tempa#1}%
  \ifx\\#2\\\expandafter\@gobble\else\expandafter\@firstofone\fi
  {\MT@in@tlist{#2}\MT@variants
   \ifMT@inlist@\else\MT@get@basefamily@#2\@nil\fi}%
}
```

`\MT@listname`
`\MT@get@listname`

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

`\MT@get@listname@`

```
\def\MT@get@listname#1{%
⟨debug⟩\MT@dinfo@nl{1}{trying to find \@nameuse{MT@abbr@#1} list for font `\MT@@font'}%
  \let\MT@listname\@undefined
  \def\@tempb{#1}%
  \MT@map@tlist@c\MT@try@order\MT@get@listname@
}
\def\MT@get@listname@#1{%
  \expandafter\MT@next@listname#1%
  \ifx\MT@listname\@undefined \else
    \expandafter\MT@tlist@break
  \fi
}
```

`\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.

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

`\MT@next@listname`

The current context is added to the font attributes. That is, the context must match.

```
\def\MT@next@listname#1#2#3#4{%
  \ifnum#1=\z@\MT@nofamilytrue\fi
  \edef\@tempa{\MT@encoding
/\ifnum#1=\@ne \MT@family  \fi
/\ifnum#2=\@ne \MT@series  \fi
/\ifnum#3=\@ne \MT@shape   \fi
/\ifnum#4=\@ne *\fi
               \MT@context}%
⟨debug⟩\MT@dinfo@nl{1}{trying \@tempa}%
  \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
    \MT@next@listname@#4%
  }{%
```

Also try with an alias family.

```
    \ifnum#1=\@ne
      \ifx\MT@familyalias\@empty \else
        \edef\@tempa{\MT@encoding
                    /\MT@familyalias
      /\ifnum#2=\@ne \MT@series\fi
      /\ifnum#3=\@ne \MT@shape\fi
      /\ifnum#4=\@ne *\fi
```

```
          \MT@context}%
⟨debug⟩\MT@dinfo@nl{1}{(alias) \@tempa}%
    \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
     \MT@next@listname@#4%
    }%
   \fi
  \fi
 }%
}
```

`\MT@next@listname@`

If size is to be evaluated, do that, otherwise use the current list.

```
\def\MT@next@listname@#1{%
 \ifnum#1=\@ne
  \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
  \ifMT@inlist@
   \let\MT@listname\MT@size@name
  \fi
 \else
  \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
 \fi
}
```

`\MT@if@list@exists`

`\MT@context`

```
\def\MT@if@list@exists{%
 \MT@let@cn\MT@context{MT@\MT@feat @context}%
 \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
 \MT@get@listname{\MT@feat @c}%
 \MT@ifdefined@c@TF\MT@listname{%
  \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
  \ifMT@nonselected
   \MT@vinfo{... Applying non-selected expansion (list `\MT@listname')}%
  \else
   \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\MT@listname'}%
  \fi
  \@firstoftwo
 }{%
```

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

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

Don't warn if `selected=false`.

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

Tracking doesn't require a list, either.

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

`\MT@get@inh@list`

`\MT@context`

The inheritance lists are global (no context).

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

 \MT@get@listname{\MT@feat @inh}%
 \MT@ifdefined@c@TF\MT@listname{%
  \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
```

```
⟨debug⟩\MT@dinfo@nl{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
⟨debug⟩                `\MT@listname'}%
    \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%
```

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

```
    \ifx\@tempc\@empty \else
⟨debug⟩\MT@dinfo@nl{1}{parsing inheritance list ...}%
```

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

```
      \begingroup
      \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
      \MT@set@inputenc{inh}%
      \expandafter\MT@inh@do\@tempc,\relax,%
      \MT@glet@nc{MT@\MT@feat @inh@\MT@listname}\@empty
      \endgroup
    \fi
  }{%
    \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
  }%
}
```

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@`

```
\def\MT@get@slot{%
  \escapechar`\\
  \let\MT@char@\m@ne
  \MT@noresttrue
```

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

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

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

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

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

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

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

- 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.

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

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

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

```
   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
     \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
  \fi
 \fi
```

```
 \let\MT@char\MT@char@
 \MT@get@slot@
 \escapechar\m@ne
}
⟨/package⟩
```

`\MT@get@slot@`

```
⟨*pdftex-def|luatex-def|xetex-def⟩
\def\MT@get@slot@{%
```

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

```
⟨xetex-def⟩  \ifnum\XeTeXfonttype\MT@font=\z@
 \ifnum\MT@char > \m@ne
```

In LuaTeX, it may also be a glyph name, prefixed with ‘/’.

```
⟨*luatex-def⟩
  \ifnum\MT@char=47\relax
   \ifMT@norest \else
    \@tempcnta=\MT@lua{
      local glyph = microtype.name_to_slot([[\expandafter\@gobble\@tempa]],true)
      if glyph then tex.write(glyph)
      else tex.write(-1)
      end
    }\relax
    \ifnum\@tempcnta<\z@
     \MT@warn@unknown
     \let\MT@char\m@ne
    \else
     \edef\MT@char{\the\@tempcnta}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}%
    \fi
   \fi
  \else
⟨/luatex-def⟩
```

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.

```
  \ifMT@norest \else
   \MT@warn@rest
⟨pdftex-def|luatex-def⟩      \let\MT@char\m@ne
⟨xetex-def⟩      \let\MT@char\@empty
  \fi
⟨luatex-def⟩    \fi
 \else
  \MT@warn@unknown
⟨xetex-def⟩    \let\MT@char\@empty
 \fi
⟨*xetex-def⟩
 \else
```

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

```
    \ifnum\MT@char=47\relax
      \ifMT@norest \edef\MT@char{U47}%
      \else
        \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
        \ifnum\@tempcnta=\z@
          \MT@warn@unknown
          \let\MT@char\@empty
        \else
          \edef\MT@char{\@tempa\space}%
          \edef\MT@char@{-\the\@tempcnta}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}%
        \fi
      \fi
    \else
      \ifnum\MT@char > \m@ne
        \ifMT@norest
```

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

```
          \@tempcnta=\XeTeXcharglyph\MT@char\relax
          \ifnum\@tempcnta=\z@
            \MT@info@missing@char
            \let\MT@char\@empty
          \else
⟨debug⟩\MT@dinfo@nl{3}{> (glyph number: \the\@tempcnta,
⟨debug⟩                   glyph name:   \XeTeXglyphname\MT@font\@tempcnta)}%
            \edef\MT@char{U\MT@char}%
          \fi
        \else
          \MT@warn@rest
          \let\MT@char\@empty
        \fi
      \else
        \MT@warn@unknown
        \let\MT@char\@empty
      \fi
    \fi
  \fi
⟨/xetex-def⟩
}
⟨/pdftex-def|luatex-def|xetex-def⟩
```

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

```
⟨*luafile⟩
if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
  local slot_of_name = luaotfload.aux.slot_of_name
  microtype.name_to_slot = function(name, unsafe)
    return slot_of_name(font.current(), name, unsafe)
  end
else
  -- we dig into internal structure (should be avoided)
  local function name_to_slot(name, unsafe)
    if fonts then
      local unicodes
      if fonts.ids then       -- legacy luaotfload
        local tfmdata = fonts.ids[font.current()]
        if not tfmdata then return end
        unicodes = tfmdata.shared.otfdata.luatex.unicodes
      else                    -- new location
        local tfmdata = fonts.hashes.identifiers[font.current()]
        if not tfmdata then return end
        unicodes = tfmdata.resources.unicodes
      end
      local unicode = unicodes[name]
      if unicode then -- does the 'or' branch actually exist?
        return type(unicode) == "number" and unicode or unicode[1]
      end
    end
  end
  microtype.name_to_slot = name_to_slot
end

⟨/luafile⟩
```

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

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

\MT@max@slot

```
⟨*pdftex-def|luatex-def|xetex-def⟩
\def\MT@max@char
⟨pdftex-def⟩  {127 }
⟨luatex-def|xetex-def⟩  {1114111 }
\def\MT@max@slot
⟨pdftex-def⟩  {255 }
⟨luatex-def|xetex-def⟩  {1114111 }
⟨/pdftex-def|luatex-def|xetex-def⟩
```

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

```
⟨*package⟩
\newif\ifMT@norest
```

```
\def\MT@is@letter#1#2\relax{%
  \ifcat a\noexpand#1\relax
    \edef\MT@char@{\number`#1}%
    \ifx\\#2\\%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a letter (\MT@char@)}%
    \else
      \MT@norestfalse
    \fi
  \else
    \ifcat !\noexpand#1\relax
      \edef\MT@char@{\number`#1}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a character (\MT@char@)}%
      \ifx\\#2\\%
        \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
      \else
        \MT@norestfalse
        \expandafter\MT@is@number#1#2\relax\relax
      \fi
    \fi
  \fi
}
```

\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.

```
\def\MT@is@number#1#2#3\relax{%
  \ifx\relax#3\relax \else
    \ifx\relax#2\relax \else
      \MT@noresttrue
      \if#1"\relax
        \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
⟨debug⟩\MT@dinfo@nl{3}{> ... a hexadecimal number: \MT@char@}%
      \else
        \if#1'\relax
          \def\MT@char@{\number#1#2#3}%
⟨debug⟩\MT@dinfo@nl{3}{> ... an octal number: \MT@char@}%
        \else
          \MT@ifint{#1#2#3}{%
            \def\MT@char@{\number#1#2#3}%
⟨debug⟩\MT@dinfo@nl{3}{> ... a decimal number: \MT@char@}%
          }\MT@norestfalse
        \fi
      \fi
      \ifnum\MT@char@ > \MT@max@slot
        \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
        \let\MT@char@\m@ne
      \fi
    \fi
  \fi
}
```

`\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.

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

Unicode handling has changed again with LaTeX 2019/10/01.

```
      \let\UTF@two@octets@noexpand\@empty
      \let\UTF@three@octets@noexpand\@empty
      \let\UTF@four@octets@noexpand\@empty
```

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

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

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

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

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

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

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

```
        \MT@toks={\the\MT@toks\space(=
                  \expandafter\expandafter\expandafter\@empty\@tempa)}%
      }%
    \x
  \fi
}
```

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

```
\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.

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

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

```
    \expandafter\expandafter\expandafter\MT@is@tlig\MT@char\relax\relax
    \ifnum\MT@char@ < \z@
```

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

```
      \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
    \fi
  \fi
}
```

`\MT@is@char`
`\MT@charstring`

A helper macro that inspects the `\meaning` of its argument.

```
\begingroup
  \catcode`\/=\z@
  /MT@map@tlist@n{/\CHARLEX}/@makeother
  /lowercase{%
    /def/x{/endgroup
      /def/MT@charstring{\CHAR"}%
      /def/MT@is@char##1\CHAR"##2##3##4/relax{%
        /ifx/relax##4/relax
          /ifMT@xunicode
            /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
              /relax/relax/relax/relax/relax
          /fi
        /else
          /ifx/relax##1/relax
            /if##3\/relax
              /edef/MT@char@{/number"##2}%
              /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
            /else
              /edef/MT@char@{/number"##2##3}%
              /MT@ifstreq/MT@charstring{##4}/relax
                {/MT@is@xchar##2##3|##4\CHAR"/relax}%
            /fi
⟨debug⟩    /MT@dinfo@nl{3}{> `/the/MT@toks' is a \char (/MT@char@)}%
          /fi
        /fi
      }%
```

`\MT@is@xchar`

With `fontspec`'s TU encoding, glyph numbers may be up to four digits.

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

`\MT@charxstring`
`\MT@strip@prefix`
`\MT@is@charx`

For `xunicode`, which doesn't `\countdef`, but rather `\def`s the chars.

```
      /def/MT@charxstring{\CHAR "}%
      /def/MT@strip@prefix##1>##2/relax{##2}%
      /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
        /ifx/relax##1/relax
          /ifx/relax##6/relax/else
            /edef/MT@char@{/number"##2##3##4##5}%
      /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@norestfalse
⟨debug⟩     /MT@dinfo@nl{3}{> `/the/MT@toks' is a xunicode \char (/MT@char@)}%
     /fi
    /fi
   }%
  }%
 }
/x
```

\MT@is@tlig

This might have to change again with the next LaTeX release, … or so I feared, but it still seems to be fine.

```
\def\MT@is@tlig#1#2{%
 \ifx#1\remove@tlig
⟨debug⟩     \MT@dinfo@nl{3}{> `\the\MT@toks' (removing remove@tlig)}%
  \MT@is@number #2\relax\relax
 \fi
}
```

\MT@is@composite

Here, we are dealing with accented characters, specified as two tokens.

```
\def\MT@is@composite#1#2\relax{%
 \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 `\string`ifying it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```
  \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
          \string\csname\MT@encoding\endcsname
          \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
```

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

```
  \ifx\UnicodeEncodingName\@undefined\else
   \expandafter\expandafter\expandafter
    \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
  \fi
  \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

Again, `xunicode`.

```
  \ifnum\MT@char@ < \z@
   \ifMT@xunicode
    \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
    \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
      \MT@char\MT@charxstring\relax\relax\relax\relax\relax
   \fi
  \fi
 \fi
}
```

\MT@is@uni@comp

Helper for `\DeclareUnicodeComposite`.

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

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

Some warning messages, for performance reasons separated here.

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

`\MT@set@listname`

```
\def\MT@set@listname{%
  \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
    `\@nameuse{MT@\MT@feat @c@name}'}%
}
```

`\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.

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

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

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

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

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

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

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

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

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

14.2.9 Hook into LaTeX's font selection

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

In contrast to the `pdfcprot` package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up.

For my reference:

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

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

```
\let\MT@font@list\@empty
\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`.

```
⟨/package⟩
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex2{
\MT@addto@setup{%
```

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

```
⟨package⟩  \MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
```

```
⟨package⟩  \MT@with@package@T{xeCJK}    {\MT@warn@unknown@once{xeCJK}}%
```

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

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

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

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

```
      \@ifpackageloaded{CJKutf8}%
        {\@ifpackagelater{CJKutf8}{2008/05/22}% 4.8.0
          {\ifpdf\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}%
          {\@firstoftwo}}%
        {\@firstoftwo}%
      {\g@addto@macro\MT@orig@pickupfont{%
        {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
           \define@newfont\else\xdef\font@name{%
             \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
      {\g@addto@macro\MT@orig@pickupfont{%
        {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
           \define@newfont\def\CJK@temp{v}%
           \ifx\CJK@temp\CJK@plane
             \expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax
             \else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi
           \else \CJK@addcmap\CJK@plane \fi
         \else\xdef\font@name{%
           \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}%
      \@gobble
    }%
  }{\@firstofone}%
```

This is the normal LaTeX definition.

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

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

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

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

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

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

```
  \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
  \g@addto@macro\pickup@font{%
      \escapechar\m@ne
⟨*package⟩
⟨debug⟩      \global\MT@inannottrue
⟨debug⟩      \MT@glet\MT@pdf@annot\@empty
⟨debug⟩      \MT@addto@annot{(line \number\inputlineno)}%
```

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

```
      \MT@let@cn\MT@font{MT@subst@\expandafter\string\font@name}%
      \ifx\MT@font\relax
        \let\MT@font\font@name
      \else
         \ifx\MT@font\font@name \else
⟨debug⟩  \MT@addto@annot{= substituted with \MT@@font}%
          \MT@register@subst@font
        \fi
      \fi
      \MT@setupfont
⟨/package⟩
⟨letterspace⟩      \MT@tracking
    \endgroup
  }%
⟨*package⟩
```

`\MT@pickupfont`
`\MT@MT@pickupfont`
`\MT@ltx@pickupfont`

Remember the patched command, because we may have to disable ourselves in certain situations.

```
  \let\MT@pickupfont\pickup@font
  \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
  \def\MT@ltx@pickupfont{\let\pickup@font\MT@orig@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.

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

`\add@accent`
`\MT@orig@add@accent`

Inside `\add@accent`, we have to disable `microtype`'s setup, since the grouping in the patched `\pickup@font` would break the accent if different fonts are used for the base character and the accent. Fortunately, LaTeX takes care that the fonts used for the `\accent` are already set up, so that we cannot be overlooking them.

```
  \let\MT@orig@add@accent\add@accent
  \def\add@accent#1#2{%
    \MT@ltx@pickupfont
    \MT@orig@add@accent{#1}{#2}%
    \MT@MT@pickupfont
  }%
⟨/package⟩
}
⟨plain⟩}\relax
⟨*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.

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

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

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

`\MT@register@subst@font` Register the substituted font (only if it isn't registered already). Additionally, we have to remove the substitute font from the list of fonts, so that we set it up again.

```
\def\MT@register@subst@font{%
  \MT@exp@one@n\MT@in@clist\font@name\MT@font@list
  \ifMT@inlist@\else
    \xdef\MT@font@list{\MT@font@list\font@name,}%
    \expandafter\MT@rem@from@clist\MT@font\MT@font@list
  \fi
}
```

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.

```
\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.

```
\def\MT@check@font@cx{%
  \MT@if@true
  \MT@map@clist@c\MT@active@features{%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
      \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \ifMT@inlist@
      \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
    \else
      \MT@if@false
    \fi
  }%
  \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
}
```

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

```
\def\MT@register@subst@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
      \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \ifMT@inlist@ \else
      \MT@exp@cs\MT@xadd
        {MT@##1@\csname MT@##1@context\endcsname font@list}%
        {\font@name,}%
      \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font
         \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \fi
  }%
}
```

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

```
\def\MT@register@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
      \MT@exp@cs\MT@xadd
```

```
        {MT@##1@\csname MT@##1@context\endcsname font@list}%
        {\MT@font,}%
      \def\@tempa{##1}%
      \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
    \fi
  }%
}
```

\MT@maybe@rem@from@list

Recurse through all context font lists of the document and remove the font, unless it's the current context.

```
\def\MT@maybe@rem@from@list#1{%
  \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
    \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
       \MT@font \csname MT@\@tempa @#1font@list\endcsname
  }%
}
```

\microtypecontext
\MT@microtypecontext

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

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

```
\def\microtypecontext{\MT@begin@catcodes\MT@microtypecontext}
\def\MT@microtypecontext#1{\MT@end@catcodes\MT@addto@setup{\microtypecontext{#1}}}
\MT@addto@setup{%
  \DeclareRobustCommand\microtypecontext{%
    \MT@begin@catcodes
    \MT@microtypecontext
  }%
  \def\MT@microtypecontext#1{%
    \MT@end@catcodes
    \MT@setup@contexts
    \let\MT@reset@context\relax
```

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

```
    \MT@glet\glb@currsize\@empty
    \setkeys{MTC}{#1}%
    \selectfont
    \MT@reset@context
  }%
}
```

\textmicrotypecontext
\MT@textmicrotypecontext
\MT@text@microtypecontext

This is just a wrapper around `\microtypecontext`.

```
\DeclareRobustCommand\textmicrotypecontext{\MT@begin@catcodes\MT@textmicrotypecontext}
\def\MT@textmicrotypecontext#1{\MT@end@catcodes\MT@text@microtypecontext{#1}}
\def\MT@text@microtypecontext#1#2{{\microtypecontext{#1}#2}}
```

\MT@reset@context
\MT@reset@context@

We have to reset the font at the end of the group, provided there actually was a change.

```
\def\MT@reset@context@{%
  \MT@vinfo{<<< Resetting contexts\on@line
⟨debug⟩  \MessageBreak= \MT@pr@context/\MT@ex@context
⟨debug⟩                /\MT@tr@context/\MT@kn@context/\MT@sp@context
  }%
  \selectfont
}
```

\MT@setup@contexts

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

```
\def\MT@setup@contexts{%
  \MT@map@clist@c\MT@active@features
    {\MT@glet@nc{MT@##1@@font@list}\MT@font@list}%
  \MT@glet\MT@check@font\MT@check@font@cx
  \MT@glet\MT@register@font\MT@register@font@cx
  \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
  \MT@glet\MT@setup@contexts\relax
}
```

Define context keys.

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

Using an empty context is only asking for trouble, therefore we choose the '`@`' instead (hoping for the LaTeX users' natural awe of this character).

```
      \MT@ifempty{##1}{\def\MT@val{@}}{\def\MT@val{##1}}%
      \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
⟨debug⟩\MT@dinfo{1}{>>> no change of #1 context: `\MT@val'}%
      \else
        \MT@vinfo{>>> Changing #1 context to `\MT@val'\MessageBreak\on@line
⟨debug⟩          \space(previous: `\@nameuse{MT@\@tempb @context}')%
                 }%
        \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

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

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

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

```
        \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
          \MT@val \csname MT@\@tempb @doc@contexts\endcsname
        \ifMT@inlist@ \else
          \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{{\MT@val}}%
⟨debug⟩  \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
        \fi
        \MT@edef@n{MT@\@tempb @context}{\MT@val}%
      \fi
    \fi
  }%
}
```

We also allow the `activate` shortcut.

```
\define@key{MTC}{activate}[]{%
  \setkeys{MTC}{protrusion={#1}}%
  \setkeys{MTC}{expansion={#1}}%
}
```

\MT@pr@context
\MT@ex@context
\MT@tr@context
\MT@sp@context
\MT@kn@context
\MT@pr@doc@contexts
\MT@ex@doc@contexts
\MT@tr@doc@contexts
\MT@sp@doc@contexts
\MT@kn@doc@contexts
\MT@extra@context

Initialise the contexts.

```
\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
  \MT@def@n{MT@#1@context}{@}%
  \MT@def@n{MT@#1@doc@contexts}{{@}}%
}
\let\MT@extra@context\@empty
```

14.3 Configuration

14.3.1 Font sets

\DeclareMicrotypeSet
\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.

```
\def\DeclareMicrotypeSet{%
```

```
  \MT@begin@catcodes
  \@ifstar
    \MT@DeclareSetAndUseIt
    \MT@DeclareSet
}
```

`\MT@DeclareSet`

```
\newcommand\MT@DeclareSet[3][]{%
  \MT@ifempty{#1}{%
    \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
  }{%
    \MT@map@clist@n{#1}{{%
      \MT@ifempty{##1}\relax{%
        \MT@is@feature{##1}{set declaration `#2'}{%
          \MT@exp@one@n\MT@declare@sets
            {\csname MT@rbba@##1\endcsname}{#2}{#3}%
        }%
      }%
    }}%
  }%
  \MT@end@catcodes
}
```

`\MT@DeclareSetAndUseIt`

```
\newcommand\MT@DeclareSetAndUseIt[3][]{%
  \MT@DeclareSet[#1]{#2}{#3}%
  \UseMicrotypeSet[#1]{#2}%
}
```

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

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

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

```
\def\MT@declare@sets#1#2#3{%
  \def\MT@curr@set@name{#2}%
  \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
    \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
    \MT@map@clist@n{font,encoding,family,series,shape,size}{%
      \MT@glet@nc{MT@#1list@##1@\MT@curr@set@name}\@undefined
    }%
  }%
  \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
⟨debug⟩\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
  \setkeys{MT@#1@set}{#3}%
}
```

`\MT@define@set@key@` ⟨*#1*⟩ = font axis, ⟨*#2*⟩ = feature.

```
\def\MT@define@set@key@#1#2{%
  \define@key{MT@#2@set}{#1}[]{%
    \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@get@highlevel{#1}%
```

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

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

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

```
    \expandafter\g@addto@macro\expandafter\MT@font@sets
      \csname MT@#2list@#1@\MT@curr@set@name\endcsname
⟨debug⟩\MT@dinfo@nl{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
  }%
```

```
}
```

`\MT@get@highlevel`

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

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

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

```
    \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
```

Test whether the command is actually defined.

```
    \MT@ifdefined@n@TF{\@tempa default}%
      {\edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}}%
      {\MT@warning{`\@backslashchar\@tempa default' is not a defined command.\MessageBreak
                   Ignoring `#1 = {\@tempa*}' in font set\MessageBreak`\MT@curr@set@name'}%
       \let\MT@val\@empty}%
```

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

```
  }%
}
```

`\MT@test@ast`

It the last character is an asterisk, execute the second argument, otherwise the first one.

```
\def\MT@test@ast#1*#2\@nil{%
  \def\@tempa{#1}%
  \MT@ifempty{#2}%
}
```

`\MT@font@sets`
`\MT@fix@font@set`

Fully expand the font specification and fix catcodes for all font sets. Also remove `fontspec`’s counters.

```
\let\MT@font@sets\@empty
\def\MT@fix@font@set#1{%
  \MT@ifdefined@c@T{#1}{%
    \xdef#1{#1}%
    \ifMT@fontspec
      \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
    \fi
    \global\@onelevel@sanitize#1%
  }%
}
```

`\MT@define@set@key@size`

`size` requires special treatment.

```
\def\MT@define@set@key@size#1{%
  \define@key{MT@#1@set}{size}[]{%
    \MT@map@clist@n{##1}{%
      \def\MT@val{####1}%
      \expandafter\MT@get@range\MT@val--\@nil
      \ifx\MT@val\relax \else
        \MT@exp@cs\MT@xadd
          {MT@#1list@size@\MT@curr@set@name}%
          {{{\MT@lower}{\MT@upper}\relax}}%
      \fi
    }%
⟨debug⟩\MT@dinfo@nl{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
  }%
}
```

Font sizes may also be specified as ranges. This has been requested by Andreas Bühmann, who has also offered valuable help in implementing this. Now, it is for instance possible to set up different lists for fonts with optical sizes. (The `MinionPro` project does this for the OpenType version of Adobe’s Minion. (Available from CTAN at `pkg/minionpro`))

\MT@get@range
\MT@upper
\MT@lower

Ranges will be stored as triplets of {⟨*lower bound*⟩} {⟨*upper bound*⟩} {⟨*list name*⟩}. For simple sizes, the upper boundary is -1.

```
\def\MT@get@range#1-#2-#3\@nil{%
  \MT@ifempty{#1}{%
    \MT@ifempty{#2}{%
      \let\MT@val\relax
    }{%
      \def\MT@lower{0}%
      \def\MT@val{#2}%
      \MT@get@size
      \edef\MT@upper{\MT@val}%
    }%
  }{%
    \def\MT@val{#1}%
    \MT@get@size
    \ifx\MT@val\relax \else
      \edef\MT@lower{\MT@val}%
      \MT@ifempty{#2}{%
        \MT@ifempty{#3}%
          {\def\MT@upper{-1}}%
```

2048 pt is TEX's maximum font size.

```
          {\def\MT@upper{2048}}%
      }{%
        \def\MT@val{#2}%
        \MT@get@size
        \ifx\MT@val\relax \else
          \MT@ifdim\MT@lower>\MT@val{%
            \MT@error{%
              Invalid size range (\MT@lower\space > \MT@val) in font set
              `\MT@curr@set@name'.\MessageBreak Swapping sizes}{}%
            \edef\MT@upper{\MT@lower}%
            \edef\MT@lower{\MT@val}%
          }{%
            \edef\MT@upper{\MT@val}%
          }%
          \MT@ifdim\MT@lower=\MT@upper
            {\def\MT@upper{-1}}%
            \relax
        \fi
      }%
    \fi
  }%
}
```

\MT@get@size

Translate a size selection command and normalise it.

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

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

```
  \if*\MT@val\relax
    \def\@tempa{\normalsize}%
  \else
    \MT@let@cn\@tempa{\MT@val}%
  \fi
  \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` instead of `\@setfontsize` because some classes might define the size selection commands by simply using `\fontsize` (e.g., the `a0poster` class).

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

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

```
  \MT@ifdimen\MT@val{%
    \@defaultunits\@tempdima\MT@val pt\relax\@nnil
    \edef\MT@val{\strip@pt\@tempdima}%
  }{%
    \MT@warning{Could not parse font size `\MT@val'\MessageBreak
                in font set `\MT@curr@set@name'}%
    \let\MT@val\relax
  }%
}
```

`\MT@define@set@key@font`

```
\def\MT@define@set@key@font#1{%
  \define@key{MT@#1@set}{font}[]{%
    \MT@glet@nc{MT@#1list@font@\MT@curr@set@name}\@empty
    \MT@map@clist@n{##1}{%
      \def\MT@val{####1}%
      \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
      \expandafter\MT@get@font\MT@val/////\@nil
      \MT@exp@two@n\g@addto@macro
        {\csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname}%
        {\MT@val,}%
    }%
    \expandafter\g@addto@macro\expandafter\MT@font@sets
      \csname MT@#1list@font@\MT@curr@set@name\endcsname
⟨debug⟩\MT@dinfo@nl{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}%
  }%
}
```

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

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

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

```
\def\MT@get@font@#1#2#3#4#5#6{%
  \let\@tempb\@empty
  \def\MT@temp{#1/#2/#3/#4/#5}%
  \MT@get@axis{encoding}{#1}%
  \MT@get@axis{family}  {#2}%
  \MT@get@axis{series}  {#3}%
  \MT@get@axis{shape}   {#4}%
  \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
  \MT@ifempty{#5}{%
    \MT@warn@axis@empty{size}{\string\normalsize}%
    \def\MT@val{*}%
  }{%
    \def\MT@val{#5}%
  }%
  \MT@get@size
}
```

`\MT@get@axis`

```
\def\MT@get@axis#1#2{%
  \def\MT@val{#2}%
  \MT@get@highlevel{#1}%
  \MT@ifempty\MT@val{%
```

```
    \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
    \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
  }\relax
  \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
}
```

`\MT@warn@axis@empty`

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

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

```
\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
  \MT@define@set@key@{encoding}{#1}%
  \MT@define@set@key@{family}  {#1}%
  \MT@define@set@key@{series}  {#1}%
  \MT@define@set@key@{shape}   {#1}%
  \MT@define@set@key@size      {#1}%
  \MT@define@set@key@font      {#1}%
}
```

`\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.

```
\def\UseMicrotypeSet{%
  \MT@begin@catcodes
  \MT@UseMicrotypeSet
}
```

`\MT@UseMicrotypeSet`

```
\newcommand*\MT@UseMicrotypeSet[2][]{%
  \MT@ifempty{#1}{%
    \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
  }{%
    \MT@map@clist@n{#1}{{%
      \MT@ifempty{##1}\relax{%
        \MT@is@feature{##1}{activation of set `#2'}{%
          \MT@exp@one@n\MT@use@set
            {\csname MT@rbba@##1\endcsname}{#2}%
        }%
      }%
    }}%
  }%
  \MT@end@catcodes
}
```

`\MT@pr@setname`
`\MT@ex@setname`
`\MT@tr@setname`
`\MT@sp@setname`
`\MT@kn@setname`
`\MT@use@set`

Only use sets that have been declared.

```
\def\MT@use@set#1#2{%
  \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
    \MT@xdef@n{MT@#1@setname}{#2}%
  }{%
    \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
      \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
    }%
    \MT@error{%
      The \@nameuse{MT@abbr@#1} set `#2' is undeclared.\MessageBreak
      Using set `\@nameuse{MT@#1@setname}' instead}{}%
  }%
}
```

`\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.

```
\def\DeclareMicrotypeSetDefault{%
```

```
  \MT@begin@catcodes
  \MT@DeclareMicrotypeSetDefault
}
```

\MT@DeclareMicrotypeSetDefault

```
\newcommand*\MT@DeclareMicrotypeSetDefault[2][]{%
  \MT@ifempty{#1}{%
    \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
  }{%
    \MT@map@clist@n{#1}{{%
      \MT@ifempty{##1}\relax{%
        \MT@is@feature{##1}{declaration of default set `#2'}{%
          \MT@exp@one@n\MT@set@default@set
            {\csname MT@rbba@##1\endcsname}{#2}%
        }%
      }%
    }}%
  }%
  \MT@end@catcodes
}
```

\MT@default@pr@set
\MT@default@ex@set
\MT@default@tr@set
\MT@default@sp@set
\MT@default@kn@set
\MT@set@default@set

```
\def\MT@set@default@set#1#2{%
  \MT@ifdefined@n@TF{MT@#1@set@@#2}{%
⟨debug⟩\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
    \MT@xdef@n{MT@default@#1@set}{#2}%
  }{%
    \MT@error{%
      The \@nameuse{MT@abbr@#1} set `#2' is not declared.\MessageBreak
      Cannot make it the default set. Using set\MessageBreak `all' instead}{}%
    \MT@xdef@n{MT@default@#1@set}{all}%
  }%
}
```

14.3.2 Variants and aliases

\DeclareMicrotypeVariants
\MT@variants

Specify suffixes for variants (see `fontname/variants.map`). The starred version appends to the list.

```
\let\MT@variants\@empty
\def\DeclareMicrotypeVariants{%
  \MT@begin@catcodes
  \@ifstar
    \MT@DeclareVariants
    {\let\MT@variants\@empty\MT@DeclareVariants}%
}
```

\MT@DeclareVariants

```
\def\MT@DeclareVariants#1{%
  \MT@map@clist@n{#1}{%
    \def\@tempa{##1}%
    \@onelevel@sanitize\@tempa
    \xdef\MT@variants{\MT@variants{\@tempa}}%
  }%
  \MT@end@catcodes
}
```

\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.

```
\def\DeclareMicrotypeAlias{%
  \MT@begin@catcodes
  \MT@DeclareMicrotypeAlias
}
```

\MT@DeclareMicrotypeAlias

```
\newcommand*\MT@DeclareMicrotypeAlias[2]{%
  \def\@tempb{#2}%
  \@onelevel@sanitize\@tempb
  \MT@ifdefined@n@T{MT@#1@alias}{%
    \MT@warning{Alias font family `\@tempb' will override
      alias `\@nameuse{MT@#1@alias}'\MessageBreak
      for font family `#1'}}%
  \MT@xdef@n{MT@#1@alias}{\@tempb}%
```

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

```
  \MT@ifdefined@c@T\MT@family{%
⟨debug⟩\MT@dinfo{1}{Activating alias font `\@tempb' for `\MT@family'}%
    \MT@glet\MT@familyalias\@tempb
  }%
  \MT@end@catcodes
}
```

\LoadMicrotypeFile

May be used to load a configuration file manually.

```
\def\LoadMicrotypeFile#1{%
  \edef\@tempa{\zap@space#1 \@empty}%
  \@onelevel@sanitize\@tempa
  \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
  \ifMT@inlist@
    \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
  \else
    \MT@xadd\MT@file@list{\@tempa,}%
    \MT@begin@catcodes
    \InputIfFileExists{mt-\@tempa.cfg}{%
      \edef\MT@curr@file{mt-\@tempa.cfg}%
      \MT@vinfo{... Loading configuration file \MT@curr@file}%
    }{%
      \MT@warning{Configuration file mt-\@tempa.cfg\MessageBreak
                  does not exist}%
    }%
    \MT@end@catcodes
  \fi
}
⟨/package⟩
⟨/package|letterspace⟩
```

14.3.3 Disabling ligatures

\DisableLigatures
\MT@DisableLigatures
\MT@nl@setname
\MT@nl@ligatures

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`'.

The optional argument may be used to disable selected ligatures only.

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex5{
\def\DisableLigatures{%
  \MT@begin@catcodes
  \MT@DisableLigatures
}
\newcommand*\MT@DisableLigatures[2][]{%
  \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
  \xdef\MT@active@features{\MT@active@features,nl}%
  \global\MT@noligaturestrue
  \MT@declare@sets{nl}{no ligatures}{#2}%
  \gdef\MT@nl@setname{no ligatures}%
  \MT@end@catcodes
}
```

```
⟨pdftex-def⟩}{
⟨/pdftex-def|luatex-def⟩
```

If pdfTEX is too old, we throw an error.

```
⟨*pdftex-def|xetex-def⟩
\renewcommand*\DisableLigatures[2][]{%
  \MT@error{Disabling ligatures of a font is only possible\MessageBreak
    with pdftex version 1.30 or newer.\MessageBreak
    Ignoring \string\DisableLigatures}{%
⟨pdftex-def⟩    Upgrade
⟨xetex-def⟩    Use
    pdftex.}%
}
⟨pdftex-def⟩}
⟨/pdftex-def|xetex-def⟩
```

14.3.4 Interaction with `babel`

`\DeclareMicrotypeBabelHook`

Declare the context that should be loaded when a `babel` language is selected. The command will not check whether a previous declaration will be overwritten.

```
⟨*package⟩
\def\DeclareMicrotypeBabelHook#1#2{%
  \MT@map@clist@n{#1}{%
    \KV@@sp@def\@tempa{##1}%
    \MT@gdef@n{MT@babel@\@tempa}{#2}%
  }%
}
⟨/package⟩
```

14.3.5 Fine tuning

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

`\SetProtrusion`

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

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

```
⟨*pdftex-def|xetex-def|luatex-def⟩
\def\SetProtrusion{%
  \MT@begin@catcodes
  \MT@SetProtrusion
}
```

`\MT@SetProtrusion`
`\MT@pr@c@name`
`\MT@extra@context`

We want the catcodes to be correct even if this is called in the preamble.

```
\newcommand*\MT@SetProtrusion[3][]{%
  \let\MT@extra@context\@empty
```

`\MT@permutelist`

Parse the optional first argument. We first have to know the name before we can deal with the extra options.

```
  \MT@set@named@keys{MT@pr@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating protrusion list `\MT@pr@c@name'}%
  \def\MT@permutelist{pr@c}%
  \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*⟩, . . .

```
  \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.

```
  \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
  \MT@end@catcodes
}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

`\SetExpansion`

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

```
⟨*pdftex-def|luatex-def⟩
\def\SetExpansion{%
  \MT@begin@catcodes
  \MT@SetExpansion
}
```

`\MT@SetExpansion`
`\MT@ex@c@name`
`\MT@extra@context`
`\MT@permutelist`

```
\newcommand*\MT@SetExpansion[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@ex@c}{#1}%
  \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @factor}{%
    \ifnum\csname MT@ex@c@\MT@ex@c@name @factor\endcsname > \@m
      \MT@warning@nl{Expansion factor \number\@nameuse{MT@ex@c@\MT@ex@c@name @factor}
        too large in list\MessageBreak `\MT@ex@c@name'. Setting it to the
        maximum of 1000}%
      \MT@glet@nc{MT@ex@c@\MT@ex@c@name @factor}\@m
    \fi
  }%
⟨debug⟩\MT@dinfo{1}{creating expansion list `\MT@ex@c@name'}%
  \def\MT@permutelist{ex@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@ex@c@\MT@ex@c@name}{#3}%
  \MT@end@catcodes
}
```

`\SetTracking`

```
\def\SetTracking{%
  \MT@begin@catcodes
  \MT@SetTracking
}
```

`\MT@SetTracking`

Third argument may be empty.

```
\newcommand*\MT@SetTracking[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@tr@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating tracking list `\MT@tr@c@name'}%
  \def\MT@permutelist{tr@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \KV@@sp@def\@tempa{#3}%
  \MT@ifempty\@tempa\relax{%
    \MT@ifint\@tempa
      {\MT@xdef@n{MT@tr@c@\MT@tr@c@name}{\@tempa}}%
      {\MT@warning{Value `\@tempa' is not a number in\MessageBreak
                   tracking set `\MT@curr@set@name'}}}%
  \MT@end@catcodes
}
⟨/pdftex-def|luatex-def⟩
```

`\SetExtraSpacing`

```
⟨*pdftex-def⟩
\def\SetExtraSpacing{%
  \MT@begin@catcodes
  \MT@SetExtraSpacing
}
```

\MT@SetExtraSpacing
\MT@sp@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExtraSpacing[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@sp@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating spacing list `\MT@sp@c@name'}%
  \def\MT@permutelist{sp@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@sp@c@\MT@sp@c@name}{#3}%
  \MT@end@catcodes
}
```

\SetExtraKerning

```
\def\SetExtraKerning{%
  \MT@begin@catcodes
  \MT@SetExtraKerning
}
```

\MT@SetExtraKerning
\MT@kn@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExtraKerning[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@kn@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating kerning list `\MT@kn@c@name'}%
  \def\MT@permutelist{kn@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@kn@c@\MT@kn@c@name}{#3}%
  \MT@end@catcodes
}
⟨/pdftex-def⟩
```

\MT@set@named@keys
\MT@options

We first set the name (if specified), then remove it from the list, and set the remaining keys.

```
⟨*package⟩
\def\MT@set@named@keys#1#2{%
  \def\x##1name=##2,##3\@nil{%
    \setkeys{#1}{name=##2}%
    \gdef\MT@options{##1##3}%
    \MT@rem@from@clist{name=}\MT@options
  }%
  \x#2,name=,\@nil
  \@expandtwoargs\setkeys{#1}\MT@options
}
```

\MT@define@code@key

Define the keys for the configuration lists (which are setting the codes, in pdfTEX speak).

```
\def\MT@define@code@key#1#2{%
  \define@key{MT@#2}{#1}[]{%
    \@tempcnta=\@ne
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
```

Here, too, we allow for something like ‘`bf*`’. It will be expanded immediately.

```
      \MT@get@highlevel{#1}%
      \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
      \advance\@tempcnta \@ne
    }%
  }%
}
```

\MT@define@code@key@family

Remove `fontspec`’s internal feature counter.

```
\def\MT@define@code@key@family#1{%
```

```
  \define@key{MT@#1}{family}[]{%
    \@tempcnta=\@ne
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@get@highlevel{family}%
      \ifMT@fontspec
        \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
      \fi
      \MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}%
      \advance\@tempcnta \@ne
    }%
  }%
}
```

`\MT@define@code@key@size`

`\MT@tempsize` must be in a `\csname`, so that it is at least `\relax`, not undefined.

```
\def\MT@define@code@key@size#1{%
  \define@key{MT@#1}{size}[]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \expandafter\MT@get@range\MT@val--\@nil
      \ifx\MT@val\relax \else
        \MT@exp@cs\MT@xadd{MT@tempsize}%
           {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
      \fi
    }%
  }%
}
```

`\MT@define@code@key@font`

```
\def\MT@define@code@key@font#1{%
  \define@key{MT@#1}{font}[]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
      \expandafter\MT@get@font@and@size\MT@val/////\@nil
      \ifMT@fontspec
        \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
      \fi
      \MT@xdef@n{MT@\MT@permutelist @\@tempb\MT@extra@context}%
        {\csname MT@\MT@permutelist @name\endcsname}%
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempb=\MT@val
⟨debug⟩               \ifx\MT@extra@context\@empty\else\MessageBreak
⟨debug⟩                 (context: \MT@extra@context)\fi}%
      \MT@exp@cs\MT@xaddb
        {MT@\MT@permutelist @\@tempb\MT@extra@context @sizes}%
        {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
    }%
  }%
}
```

`\MT@get@font@and@size`

Translate any asterisks and split off the size.

```
\def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
  \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
}
```

```
\MT@define@code@key{encoding}{cfg}
\MT@define@code@key@family   {cfg}
\MT@define@code@key{series}  {cfg}
\MT@define@code@key{shape}   {cfg}
\MT@define@code@key@size     {cfg}
\MT@define@code@key@font     {cfg}
```

`\MT@define@opt@key`

```
\def\MT@define@opt@key#1#2{%
  \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
```

```
    \MT@xdef@n{MT@#1@c@\MT@curr@set@name @#2}{##1}}}%
}
```

\MT@listname@count

The options in the optional first argument.

```
\newcount\MT@listname@count
\MT@map@clist@c\MT@features{%
```

Use file name and line number as the list name if the user didn’t bother to invent one – also check whether the name already exists (in case more than one unnamed list is loaded in the same line, for example `\AtBeginDocument`).

```
  \define@key{MT@#1@c}{name}[]{%
    \MT@ifempty{##1}{%
      \MT@ifdefined@n@TF{MT@#1@c@\MT@curr@file/\the\inputlineno}{%
        \global\advance\MT@listname@count\@ne
        \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno
                                 (\number\MT@listname@count)}%
      }{%
        \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
      }%
    }{%
      \MT@edef@n{MT@#1@c@name}{##1}%
      \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
        \MT@warning{Redefining \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'}%
      }%
    }%
    \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
  }%
  \MT@define@opt@key{#1}{load}%
  \MT@define@opt@key{#1}{factor}%
  \MT@define@opt@key{#1}{preset}%
  \MT@define@opt@key{#1}{inputenc}%
```

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

```
  \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
}
⟨/package⟩
```

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdfTeX versions, disallow. It also works with LuaTeX 0.30 or newer.

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex7{
  \define@key{MT@ex@c}{context}[]{%
    \MT@ifempty{#1}\relax{%
      \MT@glet\MT@copy@font\MT@copy@font@
      \def\MT@extra@context{#1}%
    }%
  }
  \MT@addto@setup{%
    \define@key{MT@ex@c}{context}[]{%
      \ifx\MT@copy@font\MT@copy@font@
        \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
      \else
        \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
           Ignoring `context' key\on@line}%
          {Either move the settings inside the preamble,\MessageBreak
           or load the package with the `copyfonts' option.}%
      \fi
    }%
  }
```

Protrusion contexts *might* also work without copying the font, so we don’t issue an error but only a warning. The problem is that pdfTeX only allows one set of protrusion factors for a given font within one paragraph (those that are in effect at the end of the paragraph will be in effect for the whole paragraph). When different fonts are loaded – like in the example with the footnote markers – we don't need to copy the fonts.

```
 \define@key{MT@pr@c}{context}[]{%
  \MT@ifempty{#1}\relax{%
   \MT@glet\MT@copy@font\MT@copy@font@
   \def\MT@extra@context{#1}%
  }%
 }
 \MT@addto@setup{%
  \define@key{MT@pr@c}{context}[]{%
   \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
   \ifx\MT@copy@font\MT@copy@font@\else
    \MT@warning@nl{If protrusion contexts don't work as expected,
     \MessageBreak load the package with the `copyfonts' option}%
   \fi
  }%
 }
⟨/pdftex-def|luatex-def⟩
⟨*pdftex-def⟩
}{
 \define@key{MT@ex@c}{context}[]{%
  \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
    or later. Ignoring `context' key\on@line}%
   {Upgrade pdftex.}%
 }
⟨/pdftex-def⟩
⟨*pdftex-def|xetex-def⟩
 \define@key{MT@pr@c}{context}[]{%
  \MT@error{Protrusion contexts only work with pdftex
⟨pdftex-def⟩    1.40.4\MessageBreak or later.
⟨xetex-def⟩    \MessageBreak or luatex.
    Ignoring `context' key\on@line}%
⟨pdftex-def⟩   {Upgrade pdftex.}%
⟨xetex-def⟩   {Use pdftex or luatex.}%
 }
⟨/pdftex-def|xetex-def⟩
⟨pdftex-def⟩}
```

`\MT@warn@nodim`

```
⟨*package⟩
\def\MT@warn@nodim#1{%
 \MT@warning{`\@tempa' is not a dimension.\MessageBreak
       Ignoring it and setting values relative to\MessageBreak #1}%
}
⟨/package⟩
```

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

```
⟨*pdftex-def|xetex-def|luatex-def⟩
\define@key{MT@pr@c}{unit}[character]{%
 \MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@empty
 \def\@tempa{#1}%
 \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.

```
  \MT@ifdimen\@tempa
   {\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@tempa}%
   {\MT@warn@nodim{character widths}}%
 }%
}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

Tracking may only be relative to a dimension.

```
⟨*pdftex-def|luatex-def⟩
\define@key{MT@tr@c}{unit}[1em]{%
  \MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@empty
  \def\@tempa{#1}%
  \MT@ifdimen\@tempa
    {\MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@tempa}%
    {\MT@warn@nodim{1em}%
     \MT@gdef@n{MT@tr@c@\MT@curr@set@name @unit}{1em}}%
}
⟨/pdftex-def|luatex-def⟩
```

Spacing and kerning codes may additionally be relative to space dimensions.

```
⟨*pdftex-def⟩
\MT@map@clist@n{sp,kn}{%
  \define@key{MT@#1@c}{unit}[space]{%
    \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@empty
    \def\@tempa{##1}%
    \MT@ifstreq\@tempa{character}\relax{%
      \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\m@ne
      \MT@ifstreq\@tempa{space}\relax{%
        \MT@ifdimen\@tempa
          {\MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@tempa}%
          {\MT@warn@nodim{width of space}}%
      }%
    }%
  }%
}
⟨/pdftex-def⟩
```

The first argument to `\SetExpansion` accepts some more options.

```
⟨*pdftex-def|luatex-def⟩
\MT@map@clist@n{stretch,shrink,step}{%
  \define@key{MT@ex@c}{#1}[]{%
    \MT@ifempty{##1}\relax{%
      \MT@ifint{##1}{%
```

A space terminates the number.

```
        \MT@gdef@n{MT@ex@c@\MT@curr@set@name @#1}{##1 }%
      }{%
        \MT@warning{%
          Value `##1' for option `#1' is not a number.\MessageBreak
          Ignoring it}%
      }%
    }%
  }%
}
\define@key{MT@ex@c}{auto}[true]{%
  \def\@tempa{#1}%
  \csname if\@tempa\endcsname
```

Don’t use `autoexpand` for pdfTEX version older than 1.20.

```
⟨pdftex-def⟩    \MT@requires@pdftex4%
⟨luatex-def⟩    \MT@requires@luatex3\relax
      {\MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}}%
⟨pdftex-def⟩      {\MT@warning{pdftex too old for automatic font expansion}}%
  \else
⟨pdftex-def⟩    \MT@requires@pdftex4%
⟨*luatex-def⟩
    \MT@requires@luatex3{%
      \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
                  luatex}}%
⟨/luatex-def⟩
      {\MT@glet@nc{MT@ex@c@\MT@curr@set@name @auto}\@empty}%
⟨pdftex-def⟩      \relax
  \fi
```

```
}
```

Tracking: Interword spacing and outer kerning. The variant with space just in case `\SetTracking` is called inside an argument (e.g., to `\IfFileExists`).

```
\MT@define@opt@key{tr}{spacing}
\MT@define@opt@key{tr}{outerspacing}
\MT@define@opt@key{tr}{outerkerning}
```

Which ligatures should be disabled?

```
\define@key{MT@tr@c}{noligatures}[]%
  {\MT@xdef@n{MT@tr@c@\MT@curr@set@name @noligatures}{#1}}
\define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
\define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
\define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}
⟨/pdftex-def|luatex-def⟩
```

14.3.6 Character inheritance

`\DeclareCharacterInheritance`

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

`\MT@inh@feat`
`\MT@extra@inputenc`

The optional argument may be used to restrict the list to some features, and to specify an input encoding.

```
⟨*package⟩
\renewcommand*\DeclareCharacterInheritance[1][]{%
  \let\MT@extra@context\@empty
  \let\MT@extra@inputenc\@undefined
  \let\MT@inh@feat\@empty
  \setkeys{MT@inh@}{#1}%
  \MT@begin@catcodes
  \MT@set@inh@list
}
```

`\MT@set@inh@list`

Safe category codes.

```
\def\MT@set@inh@list#1#2{%
  \MT@ifempty\MT@inh@feat{%
    \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
  }{%
    \MT@map@clist@c\MT@inh@feat{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@exp@one@n\MT@declare@char@inh
          {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
      }%
    }}%
  }%
  \MT@end@catcodes
}
```

The keys for the optional argument.

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

`\MT@declare@char@inh`

The lists cannot be given a name by the user.

```
\def\MT@declare@char@inh#1#2#3{%
  \MT@edef@n{MT@#1@inh@name}%
    {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
  \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
  \MT@ifdefined@c@T\MT@extra@inputenc{%
    \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
⟨debug⟩\MT@dinfo{1}{creating inheritance list `\@nameuse{MT@#1@inh@name}'}%
  \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
  \def\MT@permutelist{#1@inh}%
  \setkeys{MT@inh}{#2}%
  \MT@permute
}
```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations. We can reuse the key setup from the configuration lists (`\Set...`).

```
\MT@define@code@key{encoding}{inh}
\MT@define@code@key@family   {inh}
\MT@define@code@key{series}  {inh}
\MT@define@code@key{shape}   {inh}
\MT@define@code@key@size     {inh}
\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`).

```
\def\MT@inh@do#1,{%
  \ifx\relax#1\@empty \else
    \MT@inh@split #1==\relax
    \expandafter\MT@inh@do
  \fi
}
```

`\MT@inh@split`

Only gather the inheriting characters here. Their codes will actually be set in `\MT@set@`⟨*feature*⟩`@codes`.

```
⟨/package⟩
⟨*pdftex-def|xetex-def|luatex-def⟩
\def\MT@inh@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
⟨pdftex-def|luatex-def⟩    \ifnum\MT@char > \m@ne
⟨xetex-def⟩    \ifx\MT@char\@empty\else
      \let\MT@val\MT@char
      \MT@map@clist@n{#2}{%
        \def\@tempa{##1}%
        \ifx\@tempa\@empty \else
          \MT@get@slot
⟨pdftex-def|luatex-def⟩          \ifnum\MT@char > \m@ne
⟨xetex-def⟩          \ifx\MT@char\@empty\else
            \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @\MT@val @}{{\MT@char}}%
          \fi
        \fi
      }%
⟨debug⟩\MT@dinfo@nl{2}{children of #1 (\MT@val):
⟨debug⟩                 \@nameuse{MT@inh@\MT@listname @\MT@val @}}%
    \fi
  \fi
}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

14.3.7 Permutation

`\MT@permute`
`\MT@permute@`
`\MT@permute@@`
`\MT@permute@@@`
`\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. 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.

```
⟨*package⟩
\def\MT@permute{%
  \let\MT@cnt@encoding\@ne
  \MT@permute@
```

Undefine commands for the next round.

```
  \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
  \MT@glet\MT@tempsize\@undefined
}
\def\MT@permute@{%
  \let\MT@cnt@family\@ne
  \MT@permute@@
  \MT@increment\MT@cnt@encoding
  \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
    \MT@permute@
}
\def\MT@permute@@{%
  \let\MT@cnt@series\@ne
  \MT@permute@@@
  \MT@increment\MT@cnt@family
  \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
    \MT@permute@@
}
\def\MT@permute@@@{%
  \let\MT@cnt@shape\@ne
  \MT@permute@@@@
  \MT@increment\MT@cnt@series
  \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
    \MT@permute@@@
}
\def\MT@permute@@@@{%
  \MT@permute@@@@@
  \MT@increment\MT@cnt@shape
  \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
    \MT@permute@@@@
}
```

`\MT@permute@@@@@`

In order to save some memory, we can ignore unused encodings (inside the document).

```
\def\MT@permute@@@@@{%
  \MT@permute@define{encoding}%
  \ifMT@document
    \ifx\MT@tempencoding\@empty \else
      \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
        {\expandafter\expandafter\expandafter\@gobble}%
    \fi
  \fi
  \MT@permute@@@@@@
}
```

`\MT@permute@@@@@@`

```
\def\MT@permute@@@@@@{%
  \MT@permute@define{family}%
  \MT@permute@define{series}%
  \MT@permute@define{shape}%
  \edef\@tempa{\MT@tempencoding
```

```
              /\MT@tempfamily
              /\MT@tempseries
              /\MT@tempshape
              /\MT@ifdefined@c@T\MT@tempsize *}%
```

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

```
  \MT@ifstreq\@tempa{////}\relax{%
    \ifx\MT@tempencoding\@empty
      \MT@warning{%
        You have to specify an encoding for\MessageBreak
        \@nameuse{MT@abbr@\MT@permutelist} list
        `\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
        Ignoring it}%
    \else
      \MT@ifdefined@c@TF\MT@tempsize{%
```

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

```
        \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
          \MT@map@tlist@c\MT@tempsize\MT@check@rlist
        }%
        \MT@exp@cs\MT@xaddb
          {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
          \MT@tempsize
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa,\MessageBreak
⟨debug⟩         sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
⟨debug⟩                        @sizes\endcsname}%
      }{%
```

Only one list can apply to a given combination. But we don't warn if the overridden list is to be loaded by the current one.

```
        \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
          \MT@ifstreq{\csname MT@\MT@permutelist @\@tempa\MT@extra@context\endcsname}%
             {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname @load\endcsname}%
              \relax{%
            \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
              `\@nameuse{MT@\MT@permutelist @name}' will\MessageBreak override
              list `\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
              for \MessageBreak font `\@tempa'}%
          }%
        }%
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa
⟨debug⟩                \ifx\MT@extra@context\@empty\else\MessageBreak
⟨debug⟩                  (context: \MT@extra@context)\fi}%
      }%
      \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
          {\csname MT@\MT@permutelist @name\endcsname}%
    \fi
  }%
}
```

\MT@permute@define Define the commands.

```
\def\MT@permute@define#1{%
  \@tempcnta=\csname MT@cnt@#1\endcsname\relax
  \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
    {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
    {\MT@let@nc{MT@temp#1}\@empty}%
}
```

\MT@permute@reset Reset the commands.

```
\def\MT@permute@reset#1{%
  \@tempcnta=\@ne
  \MT@loop
    \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
    \advance\@tempcnta\@ne
    \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
      \iftrue
      \iffalse
  \MT@repeat
}
```

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

```
\def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

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

```
\def\MT@check@rlist@#1#2#3{%
  \def\@tempb{#1}%
  \def\@tempc{#2}%
  \MT@if@false
  \MT@exp@cs\MT@map@tlist@c
    {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
    \MT@check@range
}
```

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

```
\def\MT@check@range#1{\expandafter\MT@check@range@ #1}
```

`\MT@check@range@` `\@tempb` and `\@tempc` are lower resp. upper bound of the new range, ⟨*#1*⟩ and ⟨*#2*⟩ those of the existing range. ⟨*#3*⟩ is the list name.

```
\def\MT@check@range@#1#2#3{%
  \MT@ifdim{#2}=\m@ne{%
    \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
      \MT@ifdim\@tempb={#1}\MT@if@true\relax
    }{%
```

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

```
      \MT@ifdim\@tempb>{#1}\relax{%
        \MT@ifdim\@tempc>{#1}{%
          \MT@if@true
          \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
        }\relax
      }%
    }%
  }{%
    \MT@ifdim\@tempc=\m@ne{%
```

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

```
      \MT@ifdim\@tempb<{#2}{%
        \MT@ifdim\@tempb<{#1}\relax\MT@if@true
      }\relax
    }{%
```

- Both items are ranges.

```
      \MT@ifdim\@tempb<{#2}{%
        \MT@ifdim\@tempc>{#1}{%
          \MT@if@true
          \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
        }\relax
      }\relax
    }%
  }%
  \ifMT@if@
```

```
    \MT@ifstreq{#3}%
        {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname @load\endcsname}%
        \relax{%
      \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
        `\@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
        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.

```
    \expandafter\MT@tlist@break
  \fi
}
```

14.4 Package options

14.4.1 Declaring the options

`\ifMT@opt@expansion`
`\ifMT@opt@auto`
`\ifMT@opt@DVI`

Keep track of whether the user explicitly set these options.

```
\newif\ifMT@opt@expansion
\newif\ifMT@opt@auto
\newif\ifMT@opt@DVI
```

`\MT@optwarn@admissible`

Some warnings.

```
\def\MT@optwarn@admissible#1#2{%
  \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
                 `#2'. Assuming `false'}%
}
```

`\MT@optwarn@nan`

```
⟨/package⟩
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\def\MT@optwarn@nan#1#2{%
  \MT@warning@nl{Value `#1' for option `#2' is not a\MessageBreak number.
                 Using default value of \number\@nameuse{MT@#2@default}}%
}
⟨plain⟩}\relax
⟨/package|letterspace⟩
⟨*package⟩
```

`\MT@opt@def@set`

```
\def\MT@opt@def@set#1{%
  \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
    \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
  }{%
    \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
    \MT@warning@nl{The #1 set `\MT@val' is undeclared.\MessageBreak
                   Using set `\@nameuse{MT@\@tempb @setname}' instead}%
  }%
}
```

`expansion` and `protrusion` may be `true`, `false`, `compatibility`, `nocompatibility` and/or a ⟨*set name*⟩.

```
\MT@map@clist@n{protrusion,expansion}{%
  \define@key{MT}{#1}[true]{%
    \csname MT@opt@#1true\endcsname
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \csname MT@#1true\endcsname
        \edef\@tempb{\csname MT@rbba@#1\endcsname}%
        \MT@ifstreq\MT@val{true}\relax
        {%
          \MT@ifstreq\MT@val{false}{%
            \csname MT@#1false\endcsname
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@let@nc{MT@\@tempb @level}\@ne
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@let@nc{MT@\@tempb @level}\tw@
              }{%
```

If everything failed, it should be a set name.

```
                \MT@opt@def@set{#1}%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}
```

`activate` is a shortcut for `protrusion` and `expansion`.

```
\define@key{MT}{activate}[true]{%
   \setkeys{MT}{protrusion={#1}}%
   \setkeys{MT}{expansion={#1}}%
}
```

`spacing`, `kerning` and `tracking` do not have a compatibility level.

```
\MT@map@clist@n{spacing,kerning,tracking}{%
  \define@key{MT}{#1}[true]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \csname MT@#1true\endcsname
        \MT@ifstreq\MT@val{true}\relax
        {%
          \MT@ifstreq\MT@val{false}{%
            \csname MT@#1false\endcsname
          }{%
            \edef\@tempb{\csname MT@rbba@#1\endcsname}%
            \MT@opt@def@set{#1}%
          }%
        }%
      }%
    }%
  }%
}
```

`\MT@def@bool@opt`

The `true`/`false` options: `draft`, `final` (may be inherited from the class options), `auto`, `selected`, `babel`, `DVIoutput`, `defersetup`, `copyfonts`.

```
\def\MT@def@bool@opt#1#2{%
  \define@key{MT}{#1}[true]{%
    \def\@tempa{##1}%
    \MT@ifstreq\@tempa{true}\relax{%
      \MT@ifstreq\@tempa{false}\relax{%
        \MT@optwarn@admissible{##1}{#1}%
        \def\@tempa{false}%
      }%
    }%
    #2%
  }%
}
```

Boolean options that only set the switch.

```
\MT@map@clist@n{draft,selected,babel}{%
  \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
\MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotrue}
```

The `DVIoutput` option will change `\pdfoutput` immediately to minimise the risk of confusing other packages.

```
⟨/package⟩
⟨*pdftex-def|luatex-def|xetex-def⟩
⟨luatex-def⟩\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
\MT@def@bool@opt{DVIoutput}{%
  \csname if\@tempa\endcsname
⟨*pdftex-def|luatex-def⟩
    \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
    \pdfoutput\z@
  \else
    \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
    \pdfoutput\@ne
⟨/pdftex-def|luatex-def⟩
⟨xetex-def⟩    \MT@warning@nl{Ignoring `DVIoutput' option}%
  \fi
}
⟨/pdftex-def|luatex-def|xetex-def⟩
```

Setting the `defersetup` option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is *undocumented*, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

```
⟨*package⟩
\MT@def@bool@opt{defersetup}{%
  \csname if\@tempa\endcsname \else
    \AtEndOfPackage{%
      \MT@setup@
      \let\MT@setup@\@empty
      \let\MT@addto@setup\@firstofone
    }%
  \fi
}
⟨/package⟩
```

`copyfonts` will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This options is also *undocumented* in the hope that we can always find out automatically whether it's required. It also works with LuaTeX 0.30 or newer.

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex7{
  \MT@def@bool@opt{copyfonts}{%
    \csname if\@tempa\endcsname
      \MT@glet\MT@copy@font\MT@copy@font@
    \else
      \MT@glet\MT@copy@font\relax
    \fi
  }
⟨pdftex-def⟩}{
⟨/pdftex-def|luatex-def⟩
⟨*pdftex-def|xetex-def⟩
  \MT@def@bool@opt{copyfonts}{%
    \csname if\@tempa\endcsname
      \MT@error
⟨pdftex-def⟩       {The pdftex version you are using is too old\MessageBreak
⟨pdftex-def⟩       to use the `copyfonts' option}{Upgrade pdftex.}%
⟨xetex-def⟩        {The `copyfonts' option does not work with xetex}
⟨xetex-def⟩        {Use pdftex or luatex instead.}%
    \fi
  }
⟨pdftex-def⟩}
⟨/pdftex-def|xetex-def⟩
```

`final` is the opposite to `draft`.

```
⟨*package⟩
\MT@def@bool@opt{final}{%
  \csname if\@tempa\endcsname
    \MT@draftfalse
  \else
    \MT@drafttrue
  \fi
}
```

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

```
\define@key{MT}{verbose}[true]{%
  \let\MT@vinfo\MT@info@nl
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
    \MT@ifstreq\@tempa{errors}{%
      \let\MT@warning   \MT@warn@err
      \let\MT@warning@nl\MT@warn@err
    }{%
      \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
      \MT@ifstreq\@tempa{silent}{%
        \let\MT@warning   \MT@info
        \let\MT@warning@nl\MT@info@nl
      }{%
        \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
      }%
    }%
  }%
}
⟨/package⟩
```

Options with numerical keys: `factor`, `stretch`, `shrink`, `step`, `letterspace`.

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\MT@map@clist@n{%
⟨package⟩    stretch,shrink,step,%
    letterspace}{%
  \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
    \def\@tempa{##1 }%
```

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

```
    \MT@ifint\@tempa
      {\MT@edef@n{MT@#1}{\@tempa}}%
      {\MT@optwarn@nan{##1}{#1}}%
  }%
}
⟨plain⟩}\relax
⟨/package|letterspace⟩
```

`factor` will define the protrusion factor only.

```
⟨*package⟩
\define@key{MT}{factor}[\MT@factor@default]{%
  \def\@tempa{#1 }%
  \MT@ifint\@tempa
  {\edef\MT@pr@factor{\@tempa}}
  {\MT@optwarn@nan{#1}{factor}}%
}
```

Unit for protrusion codes.

```
\define@key{MT}{unit}[character]{%
 \def\@tempa{#1}%
 \MT@ifstreq\@tempa{character}\relax{%
  \MT@ifdimen\@tempa
   {\let\MT@pr@unit\@tempa}%
   {\MT@warning@nl{`\@tempa' is not a dimension.\MessageBreak
       Ignoring it and setting values relative to\MessageBreak
       character widths}}%
 }%
}
```

14.4.2 Loading the definition file

`\MT@endinput`

Abort if no capable engine found.

```
\let\MT@endinput\relax
```

```
\ifx\MT@engine\relax
 \MT@warning@nl{You don't seem to be using pdftex, luatex or xetex.\MessageBreak
  `\MT@MT' only works with these engines.\MessageBreak
  I will quit now}
 \MT@clear@options
\else
```

Otherwise load the engine-specific code (as strewn across this file).

```
 \input{microtype-\MT@engine tex.def}
\fi
\MT@endinput
```

14.4.3 Reading the configuration file

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as modern TEX systems have switched to the pdfTEX engine even for DVI output, so that the user might not even be aware of the fact that she's running pdfTEX.)

```
\MT@protrusiontrue
⟨/package⟩
⟨*pdftex-def|luatex-def⟩
\ifnum\pdfoutput<\@ne \else
```

Also, we only enable expansion by default if pdfTEX can expand the fonts automatically.

```
⟨pdftex-def⟩  \MT@requires@pdftex4{
  \MT@expansiontrue
⟨pdftex-def⟩    \MT@autotrue
⟨pdftex-def⟩  }\relax
\fi
⟨luatex-def⟩\MT@autotrue
⟨/pdftex-def|luatex-def⟩
```

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

The main configuration file will be loaded before processing the package options. However, the `config` option must of course be evaluated beforehand. We also have to define a no-op for the regular option processing later.

```
⟨*package⟩
\define@key{MT}{config}[]{\relax}
\def\MT@get@config#1config=#2,#3\@nil{%
```

```
  \MT@ifempty{#2}%
    {\def\MT@config@file{\MT@MT.cfg}}%
    {\def\MT@config@file{#2.cfg}}%
}
\expandafter\expandafter\expandafter\MT@get@config
  \csname opt@\@currname.\@currext\endcsname,config=,\@nil
```

Load the file.

```
\IfFileExists{\MT@config@file}{%
  \MT@info@nl{Loading configuration file \MT@config@file}%
  \MT@begin@catcodes
    \let\MT@begin@catcodes\relax
    \let\MT@end@catcodes\relax
    \let\MT@curr@file\MT@config@file
    \input{\MT@config@file}%
  \endgroup
}{\MT@warning@nl{%
    Could not find configuration file `\MT@config@file'!\MessageBreak
    This will almost certainly cause undesired results.\MessageBreak
    Please fix your installation}%
}
```

`\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).

```
\def\MT@check@active@set#1{%
  \MT@ifdefined@n@TF{MT@#1@setname}{%
    \MT@info@nl{Using \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
  }{%
    \MT@ifdefined@n@TF{MT@default@#1@set}{%
      \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
      \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
    }{%
```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '`@`', and issue a warning.

```
      \MT@gdef@n{MT@#1@setname}{@}%
      \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
                     \MessageBreak Using empty set}%
    }%
  }%
}
```

14.4.4 Hook for other packages

`\Microtype@Hook`

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

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the `microtype` package should be loaded after all font defaults have been set up (hence, using `\@ifpackageloaded` in the font package was not viable), and (2) checking `\AtBeginDocument` could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

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

```
\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\@ifpackageloaded{microtype}
  \MinionPro@MT@Hook
  {\@ifundefined{Microtype@Hook}
    {\let\Microtype@Hook\MinionPro@MT@Hook}
    {\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}}
```

`\MicroType@Hook` with a capital `T` (which only existed in version 1.7) is provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

```
\MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
  Command \string\MicroType@Hook\space is deprecated.\MessageBreak
      Use \string\Microtype@Hook\space instead}\MicroType@Hook}
\MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

14.4.5 Changing options later

`\microtypesetup`
`\MT@define@optionX`

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.

```
\def\microtypesetup{\setkeys{MT}}
\MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
⟨/package⟩
⟨*pdftex-def|luatex-def|xetex-def⟩
\def\MT@define@optionX#1#2{%
  \define@key{MTX}{#1}[true]{%
    \edef\@tempb{\csname MT@rbba@#1\endcsname}%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \@tempcnta=\m@ne
        \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.

```
          \MT@checksetup{#1}{%
            \@tempcnta=\csname MT@\@tempb @level\endcsname
            \MT@vinfo{Enabling #1
                    (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
          }%
        }{%
          \MT@ifstreq\MT@val{false}{%
            \@tempcnta=\z@
            \MT@vinfo{Disabling #1\on@line}%
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@checksetup{#1}{%
                \@tempcnta=\@ne
                \MT@let@nc{MT@\@tempb @level}\@ne
                \MT@vinfo{Setting #1 to level 1\on@line}%
              }%
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@checksetup{#1}{%
                  \@tempcnta=\tw@
                  \MT@let@nc{MT@\@tempb @level}\tw@
                  \MT@vinfo{Setting #1 to level 2\on@line}%
                }%
              }{\MT@error{Value `\MT@val' for key `#1' not recognised}
```

```
                         {Use any of `true', `false', `compatibility' or
                          `nocompatibility'.}%
              }%
            }%
          }%
        }%
        \ifnum\@tempcnta>\m@ne
          #2\@tempcnta\relax
        \fi
      }%
    }%
  }%
}
```

`\MT@checksetup`

Test whether the feature wasn't disabled in the package options.

```
\def\MT@checksetup#1{%
  \csname ifMT@#1\endcsname
    \expandafter\@firstofone
  \else
    \MT@error{You cannot enable #1 if it was disabled\MessageBreak
              in the package options}{Load microtype with #1 enabled.}%
    \expandafter\@gobble
  \fi
}
```

```
\MT@define@optionX{protrusion}\MT@protrudechars
⟨/pdftex-def|luatex-def|xetex-def⟩
⟨*pdftex-def|luatex-def⟩
\MT@define@optionX{expansion}\MT@adjustspacing
```

`\MT@protrudechars`
`\MT@adjustspacing`

```
⟨*luatex-def⟩
\MT@requires@luatex4{
  \let\pdfprotrudechars\protrudechars
  \let\pdfadjustspacing\adjustspacing
}\relax
⟨/luatex-def⟩
\let\MT@protrudechars\pdfprotrudechars
\let\MT@adjustspacing\pdfadjustspacing
⟨/pdftex-def|luatex-def⟩
⟨*xetex-def⟩
\let\MT@protrudechars\XeTeXprotrudechars
\define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
⟨/xetex-def⟩
```

`\MT@define@optionX@`

The same for `tracking`, `spacing` and `kerning`, which do not have a `compatibility` level.

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex6{
⟨luatex-def⟩\MT@requires@luatex3{
  \def\MT@define@optionX@#1#2{%
    \define@key{MTX}{#1}[true]{%
      \MT@map@clist@n{##1}{%
        \KV@@sp@def\MT@val{####1}%
        \MT@ifempty\MT@val\relax{%
          \@tempcnta=\m@ne
          \MT@ifstreq\MT@val{true}{%
            \MT@checksetup{#1}{%
              \@tempcnta=\@ne
              \MT@vinfo{Enabling #1\on@line}%
            }%
          }{%
            \MT@ifstreq\MT@val{false}{%
              \@tempcnta=\z@
              \MT@vinfo{Disabling #1\on@line}%
            }{\MT@error{Value `\MT@val' for key `#1' not recognised}
                       {Use either `true' or `false'}%
            }%
          }%
          \ifnum\@tempcnta>\m@ne
            #2\relax
          \fi
        }%
      }%
    }%
  }
```

We cannot simply let `\MT@tracking` relax, since this may select the already letter-spaced font instance.

```
  \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
                                \else \let\MT@tracking\MT@tracking@ \fi}
⟨pdftex-def⟩  \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
⟨pdftex-def⟩  \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
⟨pdftex-def⟩                               \pdfappendkern\@tempcnta}
}{
⟨/pdftex-def|luatex-def⟩
⟨*pdftex-def|luatex-def|xetex-def⟩
```

Disable for older pdfTeX versions and for XƎTeX and LuaTeX.

```
\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
⟨luatex-def⟩}
\define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
\define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
⟨pdftex-def⟩}
\define@key{MTX}{activate}[true]{%
  \setkeys{MTX}{protrusion={#1}}%
⟨pdftex-def|luatex-def⟩  \setkeys{MTX}{expansion={#1}}%
}
⟨/pdftex-def|luatex-def|xetex-def⟩
```

`\MT@saved@setupfont`

Disable everything – may be used as a temporary work-around in case setting up fonts doesn't work under certain circumstances, but only until that specific problem is fixed. This is *undocumented*, as it completely deprives us of the possibility to act – we're blind and paralysed.

```
⟨*package⟩
\let\MT@saved@setupfont\MT@setupfont

\define@key{MTX}{disable}[]{%
  \MT@info{Inactivate `\MT@MT' package}%
  \let\MT@setupfont\relax
}
\define@key{MTX}{enable}[]{%
  \MT@info{Reactivate `\MT@MT' package}%
  \let\MT@setupfont\MT@saved@setupfont
}
⟨/package⟩
```

14.4.6 Processing the options

`\MT@ProcessOptionsWithKV`

Parse options.

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\def\MT@ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\MT@temp\@empty
⟨plain⟩  \MT@requires@latex2{
    \MT@map@clist@c\@classoptionslist{%
      \def\CurrentOption{##1}%
      \MT@ifdefined@n@T{KV@#1@\expandafter\MT@getkey\CurrentOption=\@nil}{%
        \edef\MT@temp{\MT@temp,\CurrentOption,}%
        \@expandtwoargs\@removeelement\CurrentOption
          \@unusedoptionlist\@unusedoptionlist
      }%
    }%
    \edef\MT@temp{\noexpand\setkeys{#1}%
                    {\MT@temp\@ptionlist{\@currname.\@currext}}}%
```

`eplain` can handle package options.

```
⟨*plain⟩
  }{\edef\MT@temp{\noexpand\setkeys{#1}%
                    {\csname usepkg@options@\usepkg@pkg\endcsname}}}
⟨/plain⟩
  \MT@temp
  \MT@clear@options
}
```

`\MT@getkey` For `key=val` in class options.

```
\def\MT@getkey#1=#2\@nil{#1}
```

```
\MT@ProcessOptionsWithKV{MT}
⟨plain⟩}\relax
⟨/package|letterspace⟩
⟨*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).

```
\MT@addto@setup{%
\ifMT@draft
```

We disable most of what we've just defined in the 4419 lines above if we are running in draft mode.

```
  \MT@warning@nl{`draft' option active.\MessageBreak
                 Disabling all micro-typographic extensions.\MessageBreak
                 This might lead to different line and page breaks}%
  \let\MT@setupfont\relax
  \renewcommand*\LoadMicrotypeFile[1]{}%
  \renewcommand*\microtypesetup[1]{}%
  \renewcommand*\microtypecontext[1]{}%
  \renewcommand*\lsstyle{}%
\else
  \MT@setup@PDF
  \MT@setup@copies
```

Fix the font sets.

```
  \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
  \MT@setup@protrusion
  \MT@setup@expansion
  \MT@setup@tracking
  \MT@setup@warntracking
  \MT@setup@spacing
  \MT@setup@kerning
  \MT@setup@noligatures
}
⟨/package⟩
```

`\MT@setup@PDF` pdfTEX can create DVI output, too. However, both the DVI viewer and `dvips` need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

Some packages depend on the value of `\pdfoutput` and will get confused if it is changed after they have been loaded. These packages are, among others: `color`, `graphics`, `hyperref`, `crop`, `contour`, `pstricks` and, as a matter of course, `ifpdf`.

Instead of testing for each package (that's not our job), we only say that it was `microtype` that changed it. This must be sufficient!

```
⟨*pdftex-def|luatex-def⟩
\def\MT@setup@PDF{%
  \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
              \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
}
```

`\MT@setup@copies` Working on font copies?

```
\def\MT@setup@copies{%
  \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
}
⟨/pdftex-def|luatex-def⟩
⟨*xetex-def⟩
\let\MT@setup@PDF\relax
\let\MT@setup@copies\relax
⟨/xetex-def⟩
```

`\MT@setup@protrusion` Protrusion.

```
⟨*pdftex-def|xetex-def|luatex-def⟩
\def\MT@setup@protrusion{%
  \ifMT@protrusion
    \edef\MT@active@features{\MT@active@features,pr}%
    \MT@protrudechars\MT@pr@level
    \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
      \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
        factor: \number\MT@pr@factor\fi
      \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
    \MT@check@active@set{pr}%
  \else
    \let\MT@protrusion\relax
    \MT@info@nl{No character protrusion}%
  \fi
}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

`\MT@setup@expansion` For DVI output, the user must have explicitly passed the `expansion` option to the package. Under LuaTEX, expansion works quite differently: the glyphs will be positioned as if they were transformed, without actually being transformed. Since this could still be considered a viable option, we don't disable the feature completely, but issue a warning.

```
⟨*pdftex-def|luatex-def⟩
\def\MT@setup@expansion{%
  \ifnum\pdfoutput<\@ne
    \ifMT@opt@expansion
⟨*luatex-def⟩
      \ifMT@expansion
        \MT@requires@luatex3{%
          \MT@warning@nl{Font expansion doesn't work properly with luatex in\MessageBreak
             DVI mode: the glyphs won't be actually transformed,\MessageBreak
             but will only be shifted. You might want to use\MessageBreak
             pdflatex instead. I'll continue anyway ..}%
          %\MT@expansionfalse
        }\relax
      \fi
⟨/luatex-def⟩
    \else
      \MT@expansionfalse
    \fi
  \fi
  \ifMT@expansion
```

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

```
  \ifnum\MT@stretch=\m@ne
    \let\MT@stretch\MT@stretch@default
  \fi
```

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

```
  \ifnum\MT@shrink=\m@ne
    \let\MT@shrink\MT@stretch
  \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.

```
  \ifnum\MT@step=\m@ne
⟨pdftex-def⟩    \MT@requires@pdftex6{%
    \def\MT@step{1 }%
⟨*pdftex-def⟩
  }{%
    \ifnum\MT@stretch>\MT@shrink
      \ifnum\MT@shrink=\z@
        \@tempcnta=\MT@stretch
      \else
        \@tempcnta=\MT@shrink
      \fi
    \else
      \ifnum\MT@stretch=\z@
        \@tempcnta=\MT@shrink
      \else
        \@tempcnta=\MT@stretch
      \fi
    \fi
    \divide\@tempcnta 5\relax
    \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
    \edef\MT@step{\number\@tempcnta\space}%
  }%
⟨/pdftex-def⟩
  \fi
  \ifnum\MT@step=\z@
    \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
        Setting it to one}%
    \def\MT@step{1 }%
  \fi
```

`\MT@auto` Automatic expansion of the font? This new feature of pdfTEX 1.20 makes the *hz* programme really usable. It must be either '`autoexpand`' or empty (or '1000' for older versions of pdfTEX). With LuaTEX, we just leave it empty, as there's actually no difference – non-automatic font expansion doesn't work anymore. In LuaTEX 1.0.6, the '`autoexpand`' option seems to have been removed altogether and would trigger a warning.

```
  \let\MT@auto\@empty
  \ifMT@auto
```

We turn off automatic expansion if output mode is DVI.

```
⟨*pdftex-def⟩
    \MT@requires@pdftex4{%
      \ifnum\pdfoutput<\@ne
        \ifMT@opt@auto
          \MT@error{%
            Automatic font expansion only works for PDF output.\MessageBreak
            However, you are creating a DVI file}
           {If you have created expanded fonts instances, remove `auto' from%
            \MessageBreak the package options. Otherwise, you have to switch
            off expansion\MessageBreak completely.}%
          \fi
        \MT@autofalse
       \else
        \def\MT@auto{autoexpand}%
       \fi
```

Also, if pdfTEX is too old.

```
   }{%
    \MT@error{%
     The pdftex version you are using is too old for\MessageBreak
     automatic font expansion}%
    {If you have created expanded fonts instances, remove `auto' from\MessageBreak
     the package options. Otherwise, you have to switch off expansion\MessageBreak
     completely, or upgrade pdftex to version 1.20 or newer.}%
    \MT@autofalse
    \def\MT@auto{1000 }%
   }%
⟨/pdftex-def⟩
⟨luatex-def⟩      \MT@requires@luatex3\relax{\def\MT@auto{autoexpand}}%
  \else
⟨*pdftex-def⟩
```

No automatic expansion.

```
   \MT@requires@pdftex4\relax{%
    \def\MT@auto{1000 }%
   }%
⟨/pdftex-def⟩
⟨*luatex-def⟩
   \MT@requires@luatex3{%
    \ifMT@opt@auto
     \MT@error{Non-automatic font expansion does not work with\MessageBreak
          luatex}{Remove `auto=false' from the package options, or use pdftex.}%
     \MT@autotrue
    \fi
   }\relax
⟨/luatex-def⟩
  \fi
```

Choose the appropriate macro for selected expansion.

```
  \ifMT@selected
   \let\MT@set@ex@codes\MT@set@ex@codes@s
  \else
   \let\MT@set@ex@codes\MT@set@ex@codes@n
  \fi
```

Filter out `stretch=0,shrink=0`, since it would result in a pdfTEX error.

```
  \ifnum\MT@stretch=\z@
   \ifnum\MT@shrink=\z@
    \MT@warning@nl{%
     Both the stretch and shrink limit are set to zero.\MessageBreak
     Disabling font expansion}%
    \MT@expansionfalse
   \fi
  \fi
 \fi
 \ifMT@expansion
  \edef\MT@active@features{\MT@active@features,ex}%
  \MT@adjustspacing\MT@ex@level
  \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
        (level \number\MT@ex@level),\MessageBreak
        stretch: \number\MT@stretch, shrink: \number\MT@shrink,
        step: \number\MT@step, \ifMT@selected\else non-\fi selected}%
```

`\MT@check@step`

Check whether `stretch` and `shrink` are multiples of `step`.

```
    \def\MT@check@step##1{%
      \@tempcnta=\csname MT@##1\endcsname
      \divide\@tempcnta \MT@step
      \multiply\@tempcnta \MT@step
      \ifnum\@tempcnta=\csname MT@##1\endcsname\else
        \MT@warning@nl{The ##1 amount is not a multiple of step.\MessageBreak
                       The effective maximum ##1 is \the\@tempcnta\space
                       (step \number\MT@step)}%
      \fi
    }%
    \MT@check@step{stretch}%
    \MT@check@step{shrink}%
    \MT@check@active@set{ex}%
```

`\showhyphens`

Inside `\showhyphens`, font expansion should be disabled. (Since 2017/01/10, the LaTeX format contains a different version for XeTeX, but since expansion doesn't work with XeTeX, we don't have to bother.) Since 2019/10/01, the command is robust.

```
    \MT@ifdefined@n@TF{showhyphens }{%
      \def\MT@temp##1##2{%
        \expandafter\CheckCommand\csname showhyphens \endcsname[1]{##1}%
        \DeclareRobustCommand\showhyphens[1]{##2}}%
    }{%
      \def\MT@temp##1##2{%
        \CheckCommand*\showhyphens[1]{##1}%
        \gdef\showhyphens####1{##2}}%
    }%
    \MT@temp
       {\setbox0\vbox{\color@begingroup
        \everypar{}\parfillskip\z@skip
        \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
        \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}
       {\setbox0\vbox{\color@begingroup\pdfadjustspacing\z@
        \everypar{}\parfillskip\z@skip
        \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
        \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%

  \else
    \let\MT@expansion\relax
    \MT@info@nl{No font expansion}%
  \fi
}
⟨/pdftex-def|luatex-def⟩
⟨*xetex-def⟩
\def\MT@setup@expansion{%
  \ifMT@expansion
    \ifMT@opt@expansion
      \MT@error{Font expansion does not work with xetex}
               {Use pdftex or luatex instead.}%
    \fi
  \fi
}
⟨/xetex-def⟩
```

`\MT@setup@tracking`

Tracking, spacing and kerning.

```
⟨*pdftex-def|luatex-def⟩
⟨pdftex-def⟩\MT@requires@pdftex6{%
⟨luatex-def⟩\MT@requires@luatex3{%
  \def\MT@setup@tracking{%
    \ifMT@tracking
      \edef\MT@active@features{\MT@active@features,tr}%
      \MT@info@nl{Tracking enabled}%
      \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
      \ifMT@protrusion\else\MT@protrudechars\@ne\fi
    \else
      \let\MT@tracking\relax
      \MT@info@nl{No adjustment of tracking}%
    \fi
  }
⟨/pdftex-def|luatex-def⟩
```

`\MT@setup@spacing`

```
⟨*pdftex-def⟩
  \def\MT@setup@spacing{%
    \ifMT@spacing
      \edef\MT@active@features{\MT@active@features,sp}%
      \pdfadjustinterwordglue\@ne
      \MT@info@nl{Adjustment of interword spacing enabled}%
```

The `ragged2e` package sets interword spaces to a fixed value without glue. `microtype`'s modifications can therefore have undesired effects. Therefore, we issue a warning.

```
      \MT@with@package@T{ragged2e}{%
        \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
          Adjustment of interword spacing may lead to\MessageBreak
          undesired results when used with `ragged2e'.\MessageBreak
          In this case, disable the `spacing' option}%
      }%
      \MT@check@active@set{sp}%
    \else
      \let\MT@spacing\relax
      \MT@info@nl{No adjustment of interword spacing}%
    \fi
  }
```

`\MT@setup@spacing@check`

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue` > 0. Why 1500? Because some packages redefine `\frenchspacing`.[16]

```
  \def\MT@setup@spacing@check{%
    \ifMT@spacing
      \ifMT@babel \else
        \ifnum\sfcode`\. > 1500
          \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
            \MT@warning@nl{%
              \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
              interword spacing will disable it. You might want\MessageBreak
              to add `\@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
              to your preamble}%
          }%
        \fi
      \fi
    \fi
  }
```

`\MT@setup@kerning`

```
  \def\MT@setup@kerning{%
    \ifMT@kerning
      \edef\MT@active@features{\MT@active@features,kn}%
      \pdfprependkern\@ne
      \pdfappendkern\@ne
      \MT@info@nl{Adjustment of character kerning enabled}%
      \MT@check@active@set{kn}%
    \else
      \let\MT@kerning\relax
```

---

[16] Cf. the `c.t.t.` thread '`\frenchspacing` with AMS packages and babel', started by Philipp Lehman on 16 August 2005, MID: `ddtbaj$rob$1@online.de`

```
      \MT@info@nl{No adjustment of character kerning}%
    \fi
  }
⟨/pdftex-def⟩
```

\MT@error@doesnt@work

If pdfTEX is too old, we disable tracking, spacing and kerning, and throw an error message. We also switch the features off for LuaTEX and XƎTEX.

```
⟨pdftex-def|luatex-def⟩}{
⟨*luatex-def⟩
  \def\MT@setup@tracking{%
    \ifMT@tracking
      \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
        or newer. Switching it off}{Upgrade luatex.}%
      \MT@trackingfalse
      \MT@let@nc{MT@tracking}\relax
    \else
      \MT@info@nl{No adjustment of tracking (luatex too old)}%
    \fi
  }
}
⟨/luatex-def⟩
⟨*pdftex-def|xetex-def|luatex-def⟩
  \def\MT@error@doesnt@work#1{%
    \csname ifMT@#1\endcsname
      \MT@error{The #1 feature only works with pdftex 1.40\MessageBreak
        or newer. Switching it off}
⟨pdftex-def⟩       {Upgrade pdftex.}%
⟨luatex-def|xetex-def⟩       {Use pdftex instead.}%
      \csname MT@#1false\endcsname
      \MT@let@nc{MT@#1}\relax
    \else
      \MT@info@nl{No adjustment of #1%
⟨pdftex-def⟩     \space(pdftex too old)%
      }%
    \fi
  }
⟨pdftex-def|xetex-def⟩  \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
  \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
  \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
⟨pdftex-def⟩}
⟨/pdftex-def|xetex-def|luatex-def⟩
```

\MT@setup@warntracking

```
⟨letterspace⟩\MT@addto@setup
⟨pdftex-def|luatex-def⟩\def\MT@setup@warntracking
```

\MT@warn@tracking@DVI

With pdfTEX, we issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```
⟨*pdftex-def|luatex-def|letterspace⟩
{%
⟨*pdftex-def|letterspace⟩
  \ifnum\pdfoutput<\@ne
    \def\MT@warn@tracking@DVI{%
⟨letterspace⟩     \MT@pdf@or@lua{%
      \MT@warning@nl{%
          You are using tracking/letterspacing in DVI mode.\MessageBreak
          This will probably not work, unless the post-\MessageBreak
          processing program (dvips, dvipdfm(x), ...) is\MessageBreak
          able to create the virtual fonts on the fly}%
⟨letterspace⟩     }\relax
      \MT@glet\MT@warn@tracking@DVI\relax
    }%
  \else
⟨/pdftex-def|letterspace⟩
    \def\MT@warn@tracking@DVI{%
      \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
      \MT@glet\MT@warn@tracking@DVI\relax
    }%
⟨pdftex-def|letterspace⟩  \fi
  \ifnum\MT@letterspace=\m@ne
    \let\MT@letterspace\MT@letterspace@default
  \else
    \MT@ls@too@large\MT@letterspace
  \fi
}
⟨/pdftex-def|luatex-def|letterspace⟩
⟨xetex-def⟩\let\MT@setup@warntracking\relax
```

`\MT@setup@noligatures`

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```
⟨*pdftex-def|luatex-def⟩
\def\MT@setup@noligatures{%
⟨pdftex-def⟩  \MT@requires@pdftex5{%
    \ifMT@noligatures \else
      \let\MT@noligatures\relax
    \fi
⟨pdftex-def⟩  }\relax
}
⟨/pdftex-def|luatex-def⟩
⟨xetex-def⟩\let\MT@setup@noligatures\relax
```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```
⟨*package⟩
\MT@addto@setup{%
  \ifx\MT@active@features\@empty \else
    \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
  \fi
  \MT@documenttrue
}
```

`\MT@set@babel@context`

Interaction with `babel`.

```
\def\MT@set@babel@context#1{%
  \MT@ifdefined@n@TF{MT@babel@#1}{%
    \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
    \expandafter\MT@exp@one@n\expandafter\microtypecontext
      \csname MT@babel@#1\endcsname
  }{%
    \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
  }%
}
```

`\MT@shorthandoff`

Active characters can only be switched off if `babel` isn’t loaded after `microtype`.

```
\@ifpackageloaded{babel}{
  \def\MT@shorthandoff#1#2{%
    \MT@info@nl{Switching off #1 babel's active characters (#2)}%
    \shorthandoff{#2}}
}{
  \def\MT@shorthandoff#1#2{%
    \MT@error{You must load `babel' before `\MT@MT'}
             {Otherwise, `\MT@MT' cannot switch off #1 babel's\MessageBreak
              active characters.}}
}
```

We patch the language switching commands to enable language-dependent setup.

```
\MT@addto@setup{%
  \ifMT@babel
```

```
    \@ifpackageloaded{babel}{%
      \MT@info@nl{Redefining babel's language switching commands}%
      \let\MT@orig@select@language\select@language
      \def\select@language#1{%
        \MT@orig@select@language{#1}%
        \MT@set@babel@context{#1}%
      }%
      \let\MT@orig@foreign@language\foreign@language
      \def\foreign@language#1{%
        \MT@orig@foreign@language{#1}%
        \MT@set@babel@context{#1}%
      }%
      \ifMT@kerning
```

Disable French `babel`'s active characters.

```
        \MT@if@false
        \MT@with@babel@and@T{french}  \MT@if@true
        \MT@with@babel@and@T{frenchb} \MT@if@true
        \MT@with@babel@and@T{francais}\MT@if@true
        \MT@with@babel@and@T{canadien}\MT@if@true
        \MT@with@babel@and@T{acadian} \MT@if@true
        \ifMT@if@\MT@shorthandoff{French}{:;!?}\fi
```

Disable Turkish `babel`'s active characters.

```
        \MT@if@false
        \MT@with@babel@and@T{turkish} \MT@if@true
        \ifMT@if@\MT@shorthandoff{Turkish}{:!=}\fi
      \fi
```

In case `babel` was loaded before `microtype`:

```
      \MT@set@babel@context\languagename
    }{%
      \MT@warning@nl{You did not load the babel package.\MessageBreak
        The `babel' option won't have any effect}%
    }%
  \fi
}
```

Now we close the `\fi` from `\ifMT@draft`.

```
\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.

```
  \selectfont}
```

`\MT@curr@file`

This is the current file (hopefully with the correct extension).

```
\edef\MT@curr@file{\jobname.tex}
⟨/package⟩
```

Finally, execute the setup macro at the end of the preamble, and empty it (the `combine` class calls it repeatedly).

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
⟨plain⟩}\relax
⟨/package|letterspace⟩
```

Must come at the very, very end.

```
⟨package⟩\MT@ifdefined@c@T\MT@setup@spacing@check
⟨package⟩  {\AtBeginDocument{\MT@setup@spacing@check}}
```

Restore catcodes.

```
⟨package|letterspace⟩\MT@restore@catcodes
```

That was that.

15 Configuration files

Let’s now write the font configuration files.

```
⟨*config⟩

```

15.1 Font sets

We first declare some sets in the main configuration file.

```
⟨*m-t⟩
%%% -----------------------------------------------------------------------
%%% FONT SETS

\DeclareMicrotypeSet{all}
   { }

\DeclareMicrotypeSet{allmath}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }

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

\DeclareMicrotypeSet{allmath-nott}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
     family   = {rm*,sf*}
   }

\DeclareMicrotypeSet{alltext-nott}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     family   = {rm*,sf*}
   }

\DeclareMicrotypeSet{basicmath}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,OML,OMS},
     family   = {rm*,sf*},
     series   = {md*},
     size     = {normalsize,footnotesize,small,large}
   }

\DeclareMicrotypeSet{basictext}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU},
     family   = {rm*,sf*},
     series   = {md*},
     size     = {normalsize,footnotesize,small,large}
   }

\DeclareMicrotypeSet{smallcaps}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     shape    = {sc*,si,scit}
   }

\DeclareMicrotypeSet{footnotesize}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     size     = {-small}
   }

\DeclareMicrotypeSet{scriptsize}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     size     = {-footnotesize}
   }

\DeclareMicrotypeSet{normalfont}
   { font = */*/*/*/* }

```

The default sets.

```
%%% ----------------------------------------------------------------------
%%% DEFAULT SETS

\DeclareMicrotypeSetDefault[protrusion]{alltext}
\DeclareMicrotypeSetDefault[expansion] {alltext-nott}
\DeclareMicrotypeSetDefault[spacing]   {alltext-nott}
\DeclareMicrotypeSetDefault[kerning]   {alltext}
\DeclareMicrotypeSetDefault[tracking]  {smallcaps}

```

15.2 Font variants and aliases

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```
%%% ----------------------------------------------------------------------
%%% FONT VARIANTS AND ALIASES

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

Other candidates: `2` (proportional digits), `e` (engraved), `f` (Fraktur), `g` (small text), `h` (shadow), `l` (outline), `n` (informal), `p` (ornaments), `r` (roman), `s` (sans serif), `t` (typewriter). I've omitted them since they seem hardly be used and/or they are actually more than just a variant, i.e., they shouldn't share a file.

Fonts that are 'the same': The `fontspec` package will set `lmr` as the default font, whose declarations for `EU1`/`EU2`/`TU` encoding are in `mt-LatinModernRoman.cfg`. Since 2016/12/03, the default encoding with XeTeX and LuaTeX in the LaTeX format is `TU`, even if `fontspec` is not loaded.

```

\MT@if@false
\ifx\UnicodeEncodingName\@undefined\else
  \MT@ifstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
\fi
\ifMT@fontspec\MT@if@true\fi
\ifMT@if@
%% -- Computer/Latin Modern Roman
\DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
  \else
\DeclareMicrotypeAlias{lmr}{cmr}          % lmodern
\fi
```

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. We mustn't forget the Latin Modern math fonts.

```
\DeclareMicrotypeAlias{lmsy}{cmsy}        %   "
\DeclareMicrotypeAlias{lmm} {cmm}         %   "
\DeclareMicrotypeAlias{aer} {cmr}         % ae
\DeclareMicrotypeAlias{zer} {cmr}         % zefonts
\DeclareMicrotypeAlias{cmor}{cmr}         % eco
\DeclareMicrotypeAlias{hfor}{cmr}         % hfoldsty
```

Another, new Computer Modern extension.

```
\DeclareMicrotypeAlias{New Computer Modern}{Latin Modern Roman}
```

The packages `pxfonts` and `txfonts` fonts inherit Palatino and Times settings respectively, also the TeX Gyre fonts Pagella and Termes (formerly: `qfonts`).

```
%% -- Palatino
\DeclareMicrotypeAlias{pxr} {ppl}          % pxfonts
\DeclareMicrotypeAlias{qpl} {ppl}          % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)
```

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

```
\DeclareMicrotypeAlias{fp9x}{pplx}         % FPL Neu
\DeclareMicrotypeAlias{fp9j}{pplj}         %    "
```

The `newpx` package, a replacement for `pxfonts`.

```
\DeclareMicrotypeAlias{zpllf}{pplx}        % newpxtext
\DeclareMicrotypeAlias{zplosf}{pplj}       %    "
\DeclareMicrotypeAlias{zpltlf}{pplx}       %    "
\DeclareMicrotypeAlias{zpltosf}{pplj}      %    "
```

The `domitian` package.

```
\DeclareMicrotypeAlias{Domitian-TLF} {pplx}% domitian
\DeclareMicrotypeAlias{Domitian-TOsF}{pplj}%    "
```

The OpenType versions:

```
\DeclareMicrotypeAlias{Domitian}        {Palatino Linotype}
\DeclareMicrotypeAlias{TeX Gyre Pagella}{Palatino Linotype}
\DeclareMicrotypeAlias{Palatino LT Std} {Palatino Linotype}
\DeclareMicrotypeAlias{Palatino}        {Palatino Linotype}
\DeclareMicrotypeAlias{Asana Math}      {Palatino Linotype}
%% -- Times New Roman
\DeclareMicrotypeAlias{txr}{ptm}           % txfonts
```

The `newtx` package, a replacement for `txfonts`.

```
\DeclareMicrotypeAlias{ntxlf}{ptmx}        % newtxtext
\DeclareMicrotypeAlias{ntxosf}{ptmj}       %    "
\DeclareMicrotypeAlias{ntxtlf}{ptmx}       %    "
\DeclareMicrotypeAlias{ntxtosf}{ptmj}      %    "
```

The `tempora` package.

```
\DeclareMicrotypeAlias{Tempora-TLF}{ptmx}  % tempora
\DeclareMicrotypeAlias{Tempora-TOsF}{ptmj} %    "
\DeclareMicrotypeAlias{qtm}{ptm}           % TeX Gyre Termes (formerly: qfonts/QuasiTimes)
```

The `step` package.

```
\DeclareMicrotypeAlias{STEP-TLF}{ptmx}     % step
\DeclareMicrotypeAlias{STEP-TOsF}{ptmj}    %   "
```

The `stix` and `stix2` packages (the latter has departed a bit from being a Times clone, but still seems close enough).

```
\DeclareMicrotypeAlias{stix} {ptm}         % stix
\DeclareMicrotypeAlias{stix2}{ptm}         % stix2
```

More Times variants, to be checked: `pns`, `mns` (TimesNewRomanPS); `mnt` (TimesNewRomanMT, TimesNRSevenMT), `mtm` (TimesSmallTextMT); `pte` (TimesEuropa); `ptt` (TimesTen); TimesEighteen; TimesModernEF.

MicroPress’s Charter version (`chmath`).

```
%% -- Charter
\DeclareMicrotypeAlias{chr}{bch}           % CH Math
```

The `XCharter` package extends the Charter fonts.

```
\DeclareMicrotypeAlias{XCharter-TLF} {bch} % XCharter
\DeclareMicrotypeAlias{XCharter-TOsF}{bch} %    "
```

The `mathdesign` package provides math fonts matching Bitstream Charter and URW Garamond.

```
\DeclareMicrotypeAlias{mdbch}{bch}         % mathdesign/Charter
%% -- Garamond
\DeclareMicrotypeAlias{mdugm}{ugm}         % mathdesign/URW Garamond
```

The `garamondx` package, an extension of URW Garamond, providing small caps and oldstyle figures.

```
\DeclareMicrotypeAlias{zgmx}{ugm}          % garamondx
\DeclareMicrotypeAlias{zgmj}{ugm}          %    "
\DeclareMicrotypeAlias{zgmI}{ugm}          %    "
\DeclareMicrotypeAlias{zgmq}{ugm}          %    "
%% --
```

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

```
\DeclareMicrotypeAlias{ulg}{blg}           % URW LetterGothic -> Bitstream LetterGothic12Pitch
```

The `eulervm` package virtually extends the Euler fonts.

```
\DeclareMicrotypeAlias{zeur}{eur}          % Euler VM
\DeclareMicrotypeAlias{zeus}{eus}          %    "
```

Euro symbol fonts, to save some files.

```
\DeclareMicrotypeAlias{zpeus}   {zpeu}     % Adobe Euro sans -> serif
\DeclareMicrotypeAlias{eurosans}{zpeu}     % Adobe Euro sans -> serif
\DeclareMicrotypeAlias{euroitcs}{euroitc}  % ITC Euro sans -> serif

```

15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```
%%% -----------------------------------------------------------------------
%%% INTERACTION WITH THE `babel' PACKAGE

\DeclareMicrotypeBabelHook
   {english,UKenglish,british,USenglish,american}
   {kerning=, spacing=nonfrench}

\DeclareMicrotypeBabelHook
   {french,francais,acadian,canadien}
   {kerning=french, spacing=}

\DeclareMicrotypeBabelHook
   {turkish}
   {kerning=turkish, spacing=}

```

15.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\  :  \textbackslash
{  :  \textbraceleft
}  :  \textbraceright
^  :  \textasciicircum
%  :  \%
#  :  \#
```

Comma and equal sign must be guarded with braces (‘`{,}`’, ‘`{=}`’) to keep `keyval` happy.

Character commands are allowed as far as they have been defined in the proper LaTeX way, that is, when they have been assigned a slot in the font encoding with `\DeclareTextSymbol` or `\DeclareTextComposite`. Characters defined via `\chardef` are also possible.

Ligatures and `\mathchardef`’ed symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed, or else in combination with the ‘`inputenc`’ key.

With XeTeX or LuaTeX, in contrast, it is advisable to use the proper Unicode characters, or the font-specific glyph names prefixed with ‘/’ (cf. section 16).

15.5 Character inheritance

First the lists of inheriting characters. We only declare those characters that are the same on *both* sides, i.e., not Œ for O.

```
⟨/m-t⟩
⟨*m-t|zpeu|mvs⟩
%%% -----------------------------------------------------------------------
%%% CHARACTER INHERITANCE

⟨/m-t|zpeu|mvs⟩
⟨*m-t⟩
```

15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 (‘fi’ ligature), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
   { encoding = OT1 }
   { f = {011}, % ff
     i = {\i},
     j = {\j},
     O = {\O},
     o = {\o}
   }

```

15.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since LaTeX 2005/12/01 accessible as `\IJ`), 188 (‘ij’, `\ij`), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
   { encoding = T1 }
   { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
     a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
     C = {\'C,\c C,\v C},
     c = {\'c,\c c,\v c},
     D = {\v D,\DH},
     d = {\v d,\dj},
     E = {\`E,\'E,\^E,\"E,\k E,\v E},
     e = {\`e,\'e,\^e,\"e,\k e,\v e},
     f = {027}, % ff
```

```
    G = {\u G},
    g = {\u g},
    I = {\`I,\'I,\^I,\"I,\.I},
    i = {\`i,\'i,\^i,\"i,\i},
    j = {\j},
    L = {\L,\'L,\v L},
    l = {\l,\'l,\v l},
    N = {\'N,\~N,\v N},
    n = {\'n,\~n,\v n},
    O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
    o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
    R = {\'R,\v R},
    r = {\'r,\v r},
    S = {\'S,\c S,\v S,\SS},
    s = {\'s,\c s,\v s},
    T = {\c T,\v T},
    t = {\c t,\v t},
    U = {\`U,\'U,\^U,\"U,\H U,\r U},
    u = {\`u,\'u,\^u,\"u,\H u,\r u},
    Y = {\'Y,\"Y},
    y = {\'y,\"y},
    Z = {\'Z,\.Z,\v Z},
    z = {\'z,\.z,\v z}
```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```
%   - = {127},
  }

```

15.5.3 LY1

More characters: 008 (‘fl’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
  { encoding = LY1 }
  { A = {\`A,\'A,\^A,\~A,\"A,\r A},
    a = {\`a,\'a,\^a,\~a,\"a,\r a},
    C = {\c C},
    c = {\c c},
    D = {\DH},
    E = {\`E,\'E,\^E,\"E},
    e = {\`e,\'e,\^e,\"e},
    f = {011}, % ff
    I = {\`I,\'I,\^I,\"I},
    i = {\`i,\'i,\^i,\"i,\i},
    L = {\L},
    l = {\l},
    N = {\~N},
    n = {\~n},
    O = {\`O,\'O,\^O,\~O,\"O,\O},
    o = {\`o,\'o,\^o,\~o,\"o,\o},
    S = {\v S},
    s = {\v s},
    U = {\`U,\'U,\^U,\"U},
    u = {\`u,\'u,\^u,\"u},
    Y = {\'Y,\"Y},
    y = {\'y,\"y},
    Z = {\v Z},
    z = {\v z}
  }

```

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.

```
\DeclareCharacterInheritance
   { encoding = OT4 }
   { A = {\k A},
     a = {\k a},
     C = {\'C},
     c = {\'c},
     E = {\k E},
     e = {\k e},
     f = {011}, % ff
     i = {\i},
     j = {\j},
     L = {\L},
     l = {\l},
     N = {\'N},
     n = {\'n},
     O = {\O,\'O},
     o = {\o,\'o},
     S = {\'S},
     s = {\'s},
     Z = {\'Z,\.Z},
     z = {\'z,\.z},
     \textquotedblleft = "FF
   }

```

15.5.5 QX

The Central European QX encoding.[17] Ligatures: 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
   { encoding = QX }
   { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
     a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
     C = {\'C,\c C},
     c = {\'c,\c c},
     D = {\DH},
     E = {\`E,\'E,\^E,\"E,\k E},
     e = {\`e,\'e,\^e,\"e,\k e},
     f = {011}, % ff
     I = {\`I,\'I,\^I,\"I,\k I},
     i = {\`i,\'i,\^i,\"i,\k i,\i},
     j = {\j},
     L = {\L},
     l = {\l},
     N = {\'N,\~N},
     n = {\'n,\~n},
     O = {\O,\`O,\'O,\^O,\~O,\"O},
     o = {\o,\`o,\'o,\^o,\~o,\"o},
```

The Romanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously[18]) been included in QX encoding. They are still kept for backwards compatibility.

```
     S = {\'S,\c S,\textcommabelow S,\v S},
     s = {\'s,\c s,\textcommabelow s,\v s},
     T = {\c T,\textcommabelow T},
```

---

[17] Contributed by *Maciej Eder*.

[18] Cf. http://tug.org/pipermail/tex-live/2008-August/017204.html

```
    t = {\c t,\textcommabelow t},
    U = {\`U,\'U,\^U,\"U,\k U},
    u = {\`u,\'u,\^u,\"u,\k u},
    Y = {\'Y,\"Y},
    y = {\'y,\"y},
    Z = {\'Z,\.Z,\v Z},
    z = {\'z,\.z,\v z},
    . = \textellipsis
  }

```

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.

```
\DeclareCharacterInheritance
  { encoding = T5 }
  { A = {\`A,\'A,\~A,\h A,\d A,\^A,\u A,
         \`\Acircumflex,\'\Acircumflex,\~\Acircumflex,\h\Acircumflex,\d\Acircumflex,
         \`\Abreve,\'\Abreve,\~\Abreve,\h\Abreve,\d\Abreve},
    a = {\`a,\'a,\~a,\h a,\d a,\^a,\u a,
         \`\acircumflex,\'\acircumflex,\~\acircumflex,\h\acircumflex,\d\acircumflex,
         \`\abreve,\'\abreve,\~\abreve,\h\abreve,\d\abreve},
    D = {\DJ},
    d = {\dj},
    E = {\`E,\'E,\~E,\h E,\d E,\^E,
         \`\Ecircumflex,\'\Ecircumflex,\~\Ecircumflex,\h\Ecircumflex,\d\Ecircumflex},
    e = {\`e,\'e,\~e,\h e,\d e,\^e,
         \`\ecircumflex,\'\ecircumflex,\~\ecircumflex,\h\ecircumflex,\d\ecircumflex},
    I = {\`I,\'I,\~I,\h I,\d I},
    i = {\`i,\'i,\~i,\h i,\d i,\i},
    O = {\`O,\'O,\~O,\h O,\d O,\^O,\horn O,
         \`\Ocircumflex,\'\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
         \`\Ohorn,\'\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},
    o = {\`o,\'o,\~o,\h o,\d o,\^o,\horn o,
         \`\ocircumflex,\'\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
         \`\ohorn,\'\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
    U = {\`U,\'U,\~U,\h U,\d U,\horn U,
         \`\Uhorn,\'\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
    u = {\`u,\'u,\~u,\h u,\d u,\horn u,
         \`\uhorn,\'\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
    Y = {\`Y,\'Y,\~Y,\h Y,\d Y},
    y = {\`y,\'y,\~y,\h y,\d y}
  }

```

15.5.7 EU1, EU2, TU

The EU1 (XƎTEX), EU2 (LuaTEX), and, since `fontspec` version 2.5, TU encodings are not well-defined in the sense that they don't contain a fixed number of glyphs, all of which must be present. OpenType fonts may contain thousands of glyphs, but we only define those that should be present in every font (basically T1). This inheritance list should be overridden by font-specific ones.

```
\DeclareCharacterInheritance
  { encoding = {EU1,EU2,TU} }
  { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
    a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
    C = {\'C,\c C,\v C},
    c = {\'c,\c c,\v c},
    D = {\v D,\DH},
    d = {\v d,\dj},
    E = {\`E,\'E,\^E,\"E,\k E,\v E},
    e = {\`e,\'e,\^e,\"e,\k e,\v e},
%     f = {/f_f}, % sometimes /f_f, sometimes /ff
    G = {\u G},
    g = {\u g},
    I = {\`I,\'I,\^I,\"I,\.I},
    i = {\`i,\'i,\^i,\"i,\i},
%     j = {\j},
    L = {\L,\'L,\v L},
    l = {\l,\'l,\v l},
    N = {\'N,\~N,\v N},
    n = {\'n,\~n,\v n},
    O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
    o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
    R = {\'R,\v R},
    r = {\'r,\v r},
    S = {\'S,\c S,\v S}, % \SS
    s = {\'s,\c s,\v s},
    T = {\c T,\v T},
    t = {\c t,\v t},
    U = {\`U,\'U,\^U,\"U,\H U,\r U},
    u = {\`u,\'u,\^u,\"u,\H u,\r u},
    Y = {\'Y,\"Y},
    y = {\'y,\"y},
    Z = {\'Z,\.Z,\v Z},
    z = {\'z,\.z,\v z}
  }

⟨/m-t⟩
```

15.5.8 Euro symbols

Make Euro symbols settings simpler.

```
⟨*zpeu⟩
\DeclareCharacterInheritance
  { encoding = U,
    family   = {zpeu,zpeus,eurosans} }
  { E = 128 }

⟨/zpeu⟩
⟨*mvs⟩
```

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.

```
\DeclareCharacterInheritance
  { encoding = {OT1,U},
    family   = mvs }
  { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm

⟨/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.

```
⟨*m-t⟩
%%% -----------------------------------------------------------------------
%%% TRACKING/LETTERSPACING

\SetTracking
  [ name         = default,
    no ligatures = {f} ]
  { encoding     = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
  { }

```

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).

```
%%% -----------------------------------------------------------------------
%%% EXPANSION

\SetExpansion
   [ name     = default     ]
   { encoding = {OT1,OT4,QX,T1,LY1} }
   {
     A = 500,     a = 700,
   \AE = 500,   \ae = 700,
     B = 700,     b = 700,
     C = 700,     c = 700,
     D = 500,     d = 700,
     E = 700,     e = 700,
     F = 700,
     G = 500,     g = 700,
     H = 700,     h = 700,
     K = 700,     k = 700,
     M = 700,     m = 700,
     N = 700,     n = 700,
     O = 500,     o = 700,
   \OE = 500,   \oe = 700,
     P = 700,     p = 700,
     Q = 500,     q = 700,
     R = 700,
     S = 700,     s = 700,
     U = 700,     u = 700,
     W = 700,     w = 700,
     Z = 700,     z = 700,
     2 = 700,
     3 = 700,
     6 = 700,
     8 = 700,
     9 = 700
   }

```

Settings for Cyrillic T2A encoding.[19]

```
\SetExpansion
   [ name     = T2A ]
   { encoding = T2A }
   {
     A = 500,     a = 700,
     B = 700,     b = 700,
     C = 700,     c = 700,
     D = 500,     d = 700,
     E = 700,     e = 700,
     F = 700,
     G = 500,     g = 700,
     H = 700,     h = 700,
     K = 700,     k = 700,
     M = 700,     m = 700,
     N = 700,     n = 700,
     O = 500,     o = 700,
```

---

[19] Contributed by *Karl Karlsson*.

```
    P = 700,     p = 700,
    Q = 500,     q = 700,
    R = 700,
    S = 700,     s = 700,
    U = 700,     u = 700,
    W = 700,     w = 700,
    Z = 700,     z = 700,
    2 = 700,
    3 = 700,
    6 = 700,
    8 = 700,
    9 = 700,
    \CYRA = 500,     \cyra = 700,
    \CYRB = 700,     \cyrb = 700,
    \CYRV = 700,     \cyrv = 700,
    \CYRG = 700,     \cyrg = 700,
    \CYRD = 700,     \cyrd = 700,
    \CYRE = 700,     \cyre = 700,
    \CYRZH = 700,    \cyrzh = 700,
    \CYRZ = 700,     \cyrz = 700,
    \CYRI = 700,     \cyri = 700,
    \CYRISHRT = 700, \cyrishrt = 700,
    \CYRK = 700,     \cyrk = 700,
    \CYRL = 700,     \cyrl = 700,
    \CYRM = 700,     \cyrm = 700,
    \CYRN = 700,     \cyrn = 700,
    \CYRO = 500,     \cyro = 700,
    \CYRP = 700,     \cyrp = 700,
    \CYRR = 700,     \cyrr = 700,
    \CYRS = 700,     \cyrs = 700,
    \CYRT = 700,     \cyrt = 700,
    \CYRU = 700,     \cyru = 700,
    \CYRF = 700,     \cyrf = 700,
    \CYRH = 700,     \cyrh = 700,
    \CYRC = 700,     \cyrc = 700,
    \CYRCH = 700,    \cyrch = 700,
    \CYRSH = 700,    \cyrsh = 700,
    \CYRSHCH = 700,  \cyrshch = 700,
    \CYRHRDSN = 700, \cyrhrdsn = 700,
    \CYRERY = 700,   \cyrery = 700,
    \CYRSFTSN = 700, \cyrsftsn = 700,
    \CYREREV = 700,  \cyrerev = 700,
    \CYRYU = 700,    \cyryu = 700,
    \CYRYA = 700,    \cyrya = 700
  }

```

T5 encoding does not contain `\AE`, `\ae`, `\OE` and `\oe`.

```
\SetExpansion
  [ name     = T5 ]
  { encoding = T5 }
  {
    A = 500,     a = 700,
    B = 700,     b = 700,
    C = 700,     c = 700,
    D = 500,     d = 700,
    E = 700,     e = 700,
    F = 700,
    G = 500,     g = 700,
    H = 700,     h = 700,
    K = 700,     k = 700,
    M = 700,     m = 700,
    N = 700,     n = 700,
    O = 500,     o = 700,
    P = 700,     p = 700,
    Q = 500,     q = 700,
    R = 700,
    S = 700,     s = 700,
    U = 700,     u = 700,
    W = 700,     w = 700,
    Z = 700,     z = 700,
    2 = 700,
    3 = 700,
    6 = 700,
    8 = 700,
    9 = 700
  }

⟨/m-t⟩
```

15.8 Character protrusion

```
%%% -----------------------------------------------------------------------
%%% PROTRUSION

```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to `microtype` notation).

```
\SetProtrusion
   [ name     = thanh ]
   { encoding = OT1 }
   {
     A = {50,50},
     F = {  ,50},
     J = {50,  },
     K = {  ,50},
     L = {  ,50},
     T = {50,50},
     V = {50,50},
     W = {50,50},
     X = {50,50},
     Y = {50,50},
     k = {  ,50},
     r = {  ,50},
     t = {  ,50},
     v = {50,50},
     w = {50,50},
     x = {50,50},
     y = {50,50},
     . = { ,700},    {,}= { ,700},
     : = { ,500},     ; = { ,500},
     ! = { ,200},     ? = { ,200},
     ( = {50,  },     ) = {  ,50},
     - = { ,700},
     \textendash        = { ,300},     \textemdash        = { ,200},
     \textquoteleft     = {700, },     \textquoteright    = { ,700},
     \textquotedblleft  = {500, },     \textquotedblright = { ,500}
   }
```

15.8.1 Normal

The default settings always use the most moderate value.

```
⟨*cfg-t⟩
\SetProtrusion
⟨m-t⟩    [ name     = default ]
```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code `bch`)

```
⟨bch⟩   [ name     = bch-default ]
```

- Bitstream Letter Gothic (`blg`)

```
⟨blg⟩   [ name     = blg-default ]
```

- Computer Modern Roman (`cmr`)

```
⟨cmr⟩   [ name     = cmr-default ]
```

- Adobe Garamond (`pad`, `padx`, `padj`)

```
⟨pad⟩   [ name     = pad-default ]
```

- Minion[20] (`pmnx`, `pmnj`)

```
⟨pmn⟩   [ name     = pmnj-default ]
```

- Palatino (`ppl`, `pplx`, `pplj`)

```
⟨ppl⟩   [ name     = ppl-default ]
```

- Times (`ptm`, `ptmx`, `ptmj`)

```
⟨ptm⟩   [ name     = ptm-default ]
```

- URW Garamond (`ugm`)

```
⟨ugm⟩   [ name     = ugm-default ]
⟨m-t|cmr|pmn⟩   { }
⟨bch|blg|pad|ugm⟩   { encoding = OT1,
⟨ppl|ptm⟩   { encoding = {OT1,OT4},
⟨bch⟩     family   = bch }
⟨blg⟩     family   = blg }
⟨pad⟩     family   = {pad,padx,padj} }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
⟨ugm⟩     family   = ugm }
   {
⟨m-t|bch|blg|cmr|pad|pmn|ppl|ptm⟩     A = {50,50},
⟨ugm⟩     A = {50,100},
⟨pad|ptm⟩   \AE = {50,  },
⟨ugm⟩   \AE = {150,50},
⟨ugm⟩     B = {  ,50},
⟨bch|pad|pmn|ugm⟩     C = {50,  },
⟨bch|pad|pmn⟩     D = {  ,50},
⟨ugm⟩     D = {  ,70},
⟨ugm⟩     E = {  ,50},
⟨m-t|bch|cmr|pad|pmn|ptm⟩     F = {  ,50},
⟨ugm⟩     F = {  ,70},
⟨bch|pad|pmn⟩     G = {50,  },
⟨ugm⟩     G = {50,50},
⟨blg⟩     I = {150,150},
⟨m-t|cmr|pad|pmn|ppl|ptm|ugm⟩     J = {50,  },
⟨bch|blg⟩     J = {100,  },
⟨!blg⟩     K = {  ,50},
⟨blg⟩     K = {50,  },
⟨m-t|bch|cmr|pad|pmn|ppl⟩     L = {  ,50},
⟨blg⟩     L = {  ,150},
⟨ptm⟩     L = {  ,80},
⟨ugm⟩     L = {  ,120},
⟨bch|pad|pmn|ugm⟩     O = {50,50},
⟨pad⟩   \OE = {50,  },
```

---

[20] Contributed by *Harald Harders* and *Karl Karlsson*.

```
⟨ugm⟩   \OE = {50,50},
⟨blg⟩     P = {  ,100},
⟨ugm⟩     P = {  ,50},
⟨bch|pad|pmn⟩     Q = {50,70},
⟨ugm⟩     Q = {50,50},
⟨bch⟩     R = {  ,50},
⟨ugm⟩     R = {  ,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     T = {50,50},
⟨blg⟩     T = {100,100},
⟨ugm⟩     T = {70,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     V = {50,50},
⟨blg|ugm⟩     V = {70,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     W = {50,50},
⟨ugm⟩     W = {70,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     X = {50,50},
⟨ugm⟩     X = {50,70},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     Y = {50,50},
⟨blg|ptm|ugm⟩     Y = {80,80},
⟨ugm⟩     Z = {50,50},
⟨blg⟩     f = {150,100},
⟨blg⟩     i = {150,150},
⟨blg⟩     j = {100,100},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     k = {  ,50},
⟨ugm⟩     k = {  ,70},
⟨blg⟩     l = {150,150},
⟨pmn⟩     l = {  ,-50},
⟨pad|ppl⟩     p = {50,50},
⟨ugm⟩     p = {  ,50},
⟨pad|ppl⟩     q = {50,  },
⟨!blg⟩     r = {  ,50},
⟨blg⟩     r = {100, 80},
⟨cmr|pad|pmn⟩     t = {  ,70},
⟨bch⟩     t = {  ,50},
⟨blg⟩     t = {150, 80},
⟨ugm⟩     t = {  ,100},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     v = {50,50},
⟨blg⟩     v = {100,100},
⟨ugm⟩     v = {50,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     w = {50,50},
⟨ugm⟩     w = {50,70},
⟨!blg⟩     x = {50,50},
⟨blg⟩     x = {100,100},
⟨m-t|bch|pad|pmn⟩     y = {  ,50},
⟨blg⟩     y = { 50,100},
⟨cmr|ppl|ptm⟩     y = {50,70},
⟨ugm⟩     y = {  ,70},

⟨cmr⟩     0 = {  ,50},
⟨m-t⟩     1 = {50,50},
⟨bch|blg|pad|ptm|ugm⟩     1 = {150,150},
⟨cmr⟩     1 = {100,200},
⟨pmn⟩     1 = {  ,50},
⟨ppl⟩     1 = {100,100},
⟨bch|cmr|pad|ugm⟩     2 = {50,50},
⟨blg⟩     2 = {  ,100},
⟨bch|pmn⟩     3 = {50,  },
⟨cmr|pad|ugm⟩     3 = {50,50},
⟨blg⟩     3 = {100,  },
⟨m-t|pad⟩     4 = {50,50},
⟨bch⟩     4 = {100,50},
⟨blg⟩     4 = {100,  },
⟨cmr|ugm⟩     4 = {70,70},
⟨pmn⟩     4 = {50,  },
⟨ptm⟩     4 = {70,  },
⟨cmr⟩     5 = {  ,50},
⟨pad⟩     5 = {50,50},
⟨bch⟩     6 = {50,  },
⟨cmr⟩     6 = {  ,50},
⟨pad⟩     6 = {50,50},
⟨m-t⟩     7 = {50,50},
⟨bch|pad|pmn|ugm⟩     7 = {50,80},
⟨blg⟩     7 = {100,100},
⟨cmr|ptm⟩     7 = {50,100},
⟨ppl⟩     7 = {  ,50},
⟨cmr⟩     8 = {  ,50},
⟨bch|pad⟩     9 = {50,50},
⟨cmr⟩     9 = {  ,50},
⟨m-t|cmr|pad|pmn|ppl|ptm|ugm⟩     . = { ,700},
⟨bch⟩     . = { ,600},
⟨blg⟩     . = {400,500},
⟨!blg⟩    {,}= { ,500},
⟨blg⟩    {,}= {300,400},
⟨m-t|cmr|pad|pmn|ppl|ptm|ugm⟩     : = { ,500},
⟨bch⟩     : = { ,400},
⟨blg⟩     : = {300,400},
⟨m-t|bch|pad|pmn|ptm⟩     ; = { ,300},
⟨blg⟩     ; = {200,300},
⟨cmr|ppl⟩     ; = { ,500},
⟨ugm⟩     ; = { ,400},
⟨!blg⟩     ! = { ,100},
⟨blg⟩     ! = {200,200},
⟨m-t|pad|pmn|ptm⟩     ? = { ,100},
⟨bch|cmr|ppl|ugm⟩     ? = { ,200},
⟨blg⟩     ? = {150,150},
⟨pmn⟩     " = {300,300},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     @ = {50,50},
⟨ptm⟩     @ = {100,100},
⟨m-t|bch|blg|cmr|pad|pmn|ppl|ptm⟩     ~ = {200,250},
⟨ugm⟩     ~ = {300,350},
⟨pad|ppl|ptm⟩     & = {50,100},
⟨ugm⟩     & = {  ,100},
⟨m-t|cmr|pad|pmn⟩    \% = {50,50},
⟨bch⟩    \% = {  ,50},
⟨ppl|ptm⟩    \% = {100,100},
⟨ugm⟩    \% = {50,100},
⟨blg⟩    \# = {100,100},
⟨m-t|ppl|ptm|ugm⟩     * = {200,200},
⟨bch|pmn⟩     * = {200,300},
⟨blg⟩     * = {150,200},
⟨cmr|pad⟩     * = {300,300},
⟨m-t|cmr|ppl|ptm⟩     + = {250,250},
⟨bch⟩     + = {150,250},
⟨pad⟩     + = {300,300},
⟨blg|pmn⟩     + = {150,200},
⟨ugm⟩     + = {250,300},
⟨blg|ugm⟩    {=}= {200,200},
⟨m-t|pad|pmn|ptm⟩     ( = {100,   },    ) = {   ,200},
⟨bch|ugm⟩     ( = {200,   },    ) = {   ,200},
⟨cmr|blg⟩     ( = {300,   },    ) = {   ,300},
⟨ppl⟩     ( = {100,   },    ) = {   ,300},
⟨bch|pmn⟩     [ = {100,   },    ] = {   ,100},
⟨blg⟩     [ = {300,100},    ] = {   ,300},

⟨m-t|pad|pmn|ptm⟩     / = {100,200},
⟨bch⟩     / = { ,200},
⟨blg⟩     / = {300,300},
⟨cmr|ppl⟩     / = {200,300},
⟨ugm⟩     / = {100,300},
⟨m-t|ptm⟩     - = {500,500},
⟨bch|cmr|ppl⟩     - = {400,500},
⟨blg⟩     - = {300,400},
⟨pad⟩     - = {300,500},
⟨pmn⟩    - = {200,400},
⟨ugm⟩    - = {500,600},
⟨blg⟩    < = {200,100},    > = {100,200},
⟨blg⟩    _ = {150,250},
⟨blg⟩    | = {250,250},
⟨m-t|pmn⟩    \textendash       = {200,200},   \textemdash        = {150,150},
⟨bch⟩    \textendash       = {200,300},   \textemdash        = {150,250},
⟨cmr⟩    \textendash       = {400,300},   \textemdash        = {300,200},
⟨pad|ppl|ptm⟩    \textendash       = {300,300},   \textemdash        = {200,200},
⟨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).

```
⟨m-t|bch|pmn⟩    \textquoteleft    = {300,400},   \textquoteright    = {300,400},
⟨blg⟩    \textquoteleft     = {400,600},   \textquoteright    = {400,600},
⟨cmr⟩    \textquoteleft    = {500,700},   \textquoteright    = {500,600},
⟨pad|ppl⟩    \textquoteleft    = {500,700},   \textquoteright    = {500,700},
⟨ptm⟩    \textquoteleft    = {500,500},   \textquoteright    = {300,500},
⟨ugm⟩    \textquoteleft    = {300,600},   \textquoteright    = {300,600},
⟨m-t|bch|pmn⟩    \textquotedblleft = {300,300},   \textquotedblright = {300,300}
⟨blg⟩    \textquotedblright = {300,400}
⟨cmr⟩    \textquotedblleft = {500,300},   \textquotedblright = {200,600}
⟨pad|ppl|ptm⟩    \textquotedblleft = {300,400},   \textquotedblright = {300,400}
⟨ugm⟩    \textquotedblleft = {400,400},   \textquotedblright = {400,400}
   }

```

Greek uppercase letters are in OT1 encoding only.

```
⟨*m-t|cmr|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = OT1-default,
⟨cmr⟩   [ name     = cmr-OT1,
⟨pmn⟩   [ name     = pmnj-OT1,
⟨m-t⟩     load     = default ]
⟨cmr⟩     load     = cmr-default ]
⟨pmn⟩     load     = pmnj-default ]
⟨m-t⟩   { encoding = OT1 }
⟨cmr⟩   { encoding = {OT1,OT4},
⟨pmn⟩   { encoding = OT1,
⟨cmr⟩     family   = cmr  }
⟨pmn⟩     family   = pmnj }
   {
⟨m-t|cmr⟩    \AE = {50,   },
⟨pmn⟩    \OE = {50,  }
⟨*cmr⟩
     "00 = {   ,150}, % \Gamma
     "01 = {100,100}, % \Delta
     "02 = { 50, 50}, % \Theta
     "03 = {100,100}, % \Lambda
     "06 = { 50, 50}, % \Sigma
     "07 = {100,100}, % \Upsilon
     "08 = { 50, 50}, % \Phi
     "09 = { 50, 50}  % \Psi
```

Remaining slots can be found in the source file.

```
⟨/cmr⟩
   }

⟨/m-t|cmr|pmn⟩
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For XeTeX (EU1) and LuaTeX (EU2) we simply use the T1 list as default (for now).

```
\SetProtrusion
```

```
⟨m-t⟩   [ name     = T1-default,
⟨bch⟩   [ name     = bch-T1,
⟨blg⟩   [ name     = blg-T1,
⟨cmr⟩   [ name     = cmr-T1,
⟨pad⟩   [ name     = pad-T1,
⟨pmn⟩   [ name     = pmnj-T1,
⟨ppl⟩   [ name     = ppl-T1,
⟨ptm⟩   [ name     = ptm-T1,
⟨ugm⟩   [ name     = ugm-T1,
⟨m-t⟩     load     = default     ]
⟨bch⟩     load     = bch-default ]
⟨blg⟩     load     = blg-default ]
⟨cmr⟩     load     = cmr-default ]
⟨pad⟩     load     = pad-default ]
⟨pmn⟩     load     = pmnj-default ]
⟨ppl⟩     load     = ppl-default ]
⟨ptm⟩     load     = ptm-default ]
⟨ugm⟩     load     = ugm-default ]
⟨m-t⟩   { encoding = {T1,LY1,EU1,EU2,TU} }
⟨bch|cmr|pad|pmn|ppl⟩   { encoding = {T1,LY1},
⟨blg|ptm|ugm⟩   { encoding = {T1},
⟨bch⟩     family   = bch }
⟨blg⟩     family   = blg }
⟨cmr⟩     family   = cmr }
⟨pad⟩     family   = {pad,padx,padj} }
⟨pmn⟩     family   = pmnj }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
⟨ugm⟩     family   = ugm }
   {
⟨m-t|cmr⟩     \AE = {50,  },
⟨bch|pmn⟩     \OE = {50,  },
⟨pmn⟩     \TH = {  ,50},
⟨blg⟩     \v L = {   ,250},
⟨blg⟩     \v d = {   ,250},
⟨blg⟩     \v l = {   ,250},
⟨blg⟩     \v t = {   ,250},
⟨blg⟩     127 =  {300,400},
⟨blg⟩     156 =  {100,   }, % IJ
⟨blg⟩     188 =  { 80, 80}, % ij
⟨m-t|bch|pad|pmn|ppl|ptm⟩       _ = {100,100},
⟨cmr⟩       _ = {200,200},
⟨ugm⟩       _ = {100,200},
⟨m-t|pad|pmn|ptm⟩     \textbackslash    = {100,200},
⟨bch⟩     \textbackslash    = {150,200},
⟨blg⟩     \textbackslash    = {250,300},
⟨cmr|ppl⟩     \textbackslash    = {200,300},
⟨ugm⟩     \textbackslash    = {100,300},
⟨ugm⟩     \textbar          = {200,200},
⟨blg⟩     \textendash       = {300,300},   \textemdash        = {150,150},
⟨blg⟩     \textquotedbl     = {300,400},   \textquotedblleft  = {300,400},
⟨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.

```
⟨m-t|cmr|pad|ppl|ptm|ugm⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨blg⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,400},
⟨bch|pmn⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨m-t|bch|pmn⟩     \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨blg⟩     \guilsinglleft    = {300,500},   \guilsinglright    = {300,500},
⟨cmr|pad|ppl|ptm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨ugm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch|pmn⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨blg|pad|ppl|ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ugm⟩     \guillemotleft    = {300,400},   \guillemotright    = {300,400},
⟨m-t|bch|cmr|pad|pmn|ppl|ugm⟩     \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨blg⟩     \textexclamdown   = {200,   },   \textquestiondown  = {100,   },
⟨ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨m-t|cmr|pad|ppl|ptm|ugm⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨bch|blg|pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t|bch|cmr|pad|ppl|ptm|ugm⟩     \textless         = {200,100},   \textgreater       = {100,200}
⟨pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100},
⟨pmn⟩     \textvisiblespace = {100,100} % not in LY1
  }

```

The `lmodern` fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```
⟨*cmr⟩
\SetProtrusion
  [ name     = lmr-T1,
    load     = cmr-T1   ]
  { encoding = {T1,LY1},
    family   = lmr      }
  {
    \textquotedblleft  = {300,400},  \textquotedblright = {300,400}
  }

⟨/cmr⟩
```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).[21]

```
⟨*m-t|cmr|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = T2A-default,
⟨cmr⟩   [ name     = cmr-T2A,
⟨pmn⟩   [ name     = pmnj-T2A,
⟨m-t⟩     load     = default     ]
⟨cmr⟩     load     = cmr-default ]
⟨pmn⟩     load     = pmnj-default ]
  { encoding = T2A,
⟨m-t⟩   }
⟨cmr⟩     family   = cmr }
⟨pmn⟩     family   = pmnj }
  {
    \CYRA = {50,50},
    \CYRG = {  ,50},
    \CYRK = {  ,50},
    \CYRT = {50,50},
    \CYRH = {50,50},
    \CYRU = {50,50},
⟨pmn⟩     \CYRS = {50,  },
⟨pmn⟩     \CYRO = {50,50},
    \cyrk = {  ,50},
    \cyrg = {  ,50},
    \cyrh = {50,50},
⟨m-t|pmn⟩     \cyru = {50,50},
⟨cmr⟩     \cyru = {50,70},
⟨m-t⟩         _  = {100,100},
⟨cmr⟩         _  = {200,200},
⟨m-t⟩     \textbackslash    = {100,200},   \quotedblbase      = {400,400},
⟨cmr⟩     \textbackslash    = {200,300},   \quotedblbase      = {400,400},
⟨pmn⟩     \textbackslash    = {100,200},   \quotedblbase      = {300,300},
⟨cmr⟩     \textquotedbl     = {300,300},   \textquotedblleft  = {200,600},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨pmn⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
```

---

[21] Contributed by *Karl Karlsson*.

```
⟨m-t|cmr⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t|cmr⟩     \textless         = {200,100},   \textgreater       = {100,200}
⟨pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100}
   }

⟨/m-t|cmr|pmn⟩
```

Settings for the QX encoding (generic and Times).[22] It also includes some glyphs otherwise in TS1.

```
⟨*m-t|ptm⟩
\SetProtrusion
⟨m-t⟩   [ name     = QX-default,
⟨ptm⟩   [ name     = ptm-QX,
⟨m-t⟩     load     = default ]
⟨ptm⟩     load     = ptm-default ]
⟨m-t⟩   { encoding = QX }
⟨ptm⟩   { encoding = QX,
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
   {
     \AE = {50,  },
⟨ptm⟩      *  = {200,200},
     {=} = {100,100},
     \textunderscore   = {100,100},
     \textbackslash    = {100,200},
     \quotedblbase     = {400,400},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
     \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨m-t⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨ptm⟩     \textbraceleft    = {200,200},   \textbraceright    = {200,300},
     \textless         = {200,100},   \textgreater       = {100,200},
     \textminus        = {200,200},   \textdegree        = {300,300},
⟨m-t⟩     \copyright        = {100,100},   \textregistered    = {100,100}
⟨ptm⟩     \copyright        = {100,150},   \textregistered    = {100,150},
⟨ptm⟩     \textxgeq         = {   ,100},   \textxleq          = {100,   },
⟨ptm⟩     \textalpha        = {   , 50},   \textDelta         = { 70, 70},
⟨ptm⟩     \textpi           = { 50, 80},   \textSigma         = {   , 70},
⟨ptm⟩     \textmu           = {   , 80},   \texteuro          = { 50, 50},
⟨ptm⟩     \textellipsis     = {150,200},   \textasciitilde    = { 80, 80},
⟨ptm⟩     \textapprox       = { 50, 50},   \textinfty         = {100,100},
⟨ptm⟩     \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
⟨ptm⟩     \textdiv          = { 50,150},   \textsection       = { 80, 80},
⟨ptm⟩     \texttimes        = {100,150},   \textpm            = { 50, 80},
⟨ptm⟩     \textbullet       = {150,150},   \textperiodcentered = {300,300},
⟨ptm⟩     \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
⟨ptm⟩     \textperthousand  = {    ,50}
   }

⟨/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.

```
⟨*cmr|bch⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-T5,
⟨cmr⟩     load     = cmr-default ]
⟨bch⟩   [ name     = bch-T5,
⟨bch⟩     load     = bch-default ]
   { encoding = T5,
⟨cmr⟩     family   = cmr }
⟨bch⟩     family   = bch }
   {
```

---

[22] Contributed by *Maciej Eder*.

```
⟨bch⟩      _ = {100,100},
⟨bch⟩      \textbackslash    = {150,200},
⟨cmr⟩      \textbackslash    = {200,300},
⟨cmr⟩      \textquotedblleft = {200,600},
⟨cmr⟩      \textquotedbl     = {300,300},
⟨bch⟩      \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨cmr⟩      \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨bch⟩      \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨cmr⟩      \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨bch⟩      \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨cmr⟩      \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch⟩      \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨cmr⟩      \textbraceleft    = {400,200},   \textbraceright    = {200,400},
     \textless         = {200,100},   \textgreater       = {100,200}
   }

⟨/cmr|bch⟩
```

Minion with lining numbers.

```
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-OT1,
     load     = pmnj-default ]
   { encoding = OT1,
     family   = pmnx }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-T1,
     load     = pmnj-T1  ]
   { encoding = {T1,LY1},
     family   = pmnx     }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-T2A,
     load     = pmnj-T2A  ]
   { encoding = {T2A},
     family   = pmnx     }
   {
     1 = {230,180}
   }

⟨/pmn⟩
```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```
⟨*ptm⟩
 \SetProtrusion
   [ name     = ptm-LY1,
     load     = ptm-T1 ]
   { encoding = LY1,
     family   = {ptm,ptmx,ptmj} }
   {
     _                        = {100,100},
     \texttrademark           = {100,100},
     \textregistered          = {100,100},
     \textcopyright           = {100,100},
     \textdegree              = {300,300},
     \textminus               = {200,200},
     \textellipsis            = {150,200},
%    \texteuro                 = {   ,   }, % ?
     \textcent                 = {100,100},
     \textquotesingle          = {500,500},
     \textflorin               = { 50, 70},
     \textdagger               = {150,150},
     \textdaggerdbl            = {100,100},
     \textperthousand          = {   , 50},
     \textbullet               = {150,150},
     \textonesuperior          = {100,100},
     \texttwosuperior          = { 50, 50},
     \textthreesuperior        = { 50, 50},
     \textperiodcentered       = {300,300},
     \textplusminus            = { 50, 80},
     \textmultiply             = {100,100},
     \textdivide               = { 50,150}
```

Remaining slots in the source file.

```
   }

⟨/ptm⟩
```

15.8.2 Italics

To find default settings for italic is difficult, since the character shapes and their behaviour at the beginning or end of line may be wildly different for different fonts. In the generic settings we therefore omit the letters, and only set up the punctuation characters.

The italic glyphs of Computer Modern Roman feature a lot of side bearing, therefore almost all of them have to protrude.[23]

```
\SetProtrusion
⟨m-t⟩   [ name     = OT1-it   ]
⟨bch⟩   [ name     = bch-it   ]
⟨blg⟩   [ name     = blg-it,
⟨blg⟩     load     = blg-default ]
⟨cmr⟩   [ name     = cmr-it   ]
⟨pad⟩   [ name     = pad-it   ]
⟨pmn⟩   [ name     = pmnj-it  ]
⟨ppl⟩   [ name     = ppl-it   ]
⟨ptm⟩   [ name     = ptm-it   ]
⟨ugm⟩   [ name     = ugm-it   ]
⟨m-t|bch|blg|pad|ugm⟩   { encoding = OT1,
⟨ppl|ptm⟩   { encoding = {OT1,OT4},
⟨bch⟩     family   = bch,
⟨blg⟩     family   = blg,
⟨pad⟩     family   = {pad,padx,padj},
⟨ppl⟩     family   = {ppl,pplx,pplj},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
⟨ugm⟩     family   = ugm,
⟨m-t|bch|pad|ppl|ptm⟩     shape    = {it,sl}  }
⟨blg|ugm⟩     shape    = it  }
⟨cmr|pmn⟩   { }
   {
⟨cmr⟩     A = {100,100},
⟨ptm⟩     A = {100,50},
⟨pad|pmn⟩     A = {50,  },
⟨ugm⟩     A = {  ,150},
⟨ppl⟩     A = {50,50},
⟨ptm⟩   \AE = {100,  },
⟨pad|ppl⟩   \AE = {50,  },
⟨cmr⟩     B = {83,-40},
```

---

[23] Settings contributed by *Hendrik Vogt*.

```
⟨pad|ppl|ptm⟩     B = {50,  },
⟨pmn⟩     B = {20,-50},
⟨bch|ppl|ptm|ugm⟩     C = {50,  },
⟨cmr⟩     C = {165,-75},
⟨pad⟩     C = {100, },
⟨pmn⟩     C = {50,-50},
⟨cmr⟩     D = {75, -28},
⟨pad|ppl|ptm⟩     D = {50,50},
⟨pmn⟩     D = {20,  },
⟨cmr⟩     E = {80,-55},
⟨pad|ppl|ptm⟩     E = {50,  },
⟨pmn⟩     E = {20,-50},
⟨cmr⟩     F = {85,-80},
⟨pad|ptm⟩     F = {100, },
⟨pmn⟩     F = {10,  },
⟨ppl⟩     F = {50,  },
⟨bch|ppl|ptm|ugm⟩     G = {50,  },
⟨cmr⟩     G = {153,-15},
⟨pad⟩     G = {100, },
⟨pmn⟩     G = {50,-50},
⟨cmr⟩     H = {73,-60},
⟨pad|ppl|ptm⟩     H = {50,  },
⟨cmr⟩     I = {140,-120},
⟨pad|ptm⟩     I = {50,  },
⟨pmn⟩     I = {20,-50},
⟨cmr⟩     J = {135,-80},
⟨pad⟩     J = {50,  },
⟨pmn⟩     J = {20,  },
⟨ptm⟩     J = {100, },
⟨cmr⟩     K = {70,-30},
⟨pad|ppl|ptm⟩     K = {50,  },
⟨pmn⟩     K = {20,  },
⟨cmr⟩     L = {87, 40},
⟨pad|ppl|ptm⟩     L = {50,  },
⟨pmn⟩     L = {20,50},
⟨ugm⟩     L = {  ,100},
⟨cmr⟩     M = {67,-45},
⟨pmn⟩     M = {  ,-30},
⟨ptm⟩     M = {50,  },
⟨cmr⟩     N = {75,-55},
⟨pmn⟩     N = {  ,-30},
⟨ptm⟩     N = {50,  },
⟨bch|pmn|ppl|ptm⟩     O = {50,  },
⟨cmr⟩     O = {150,-30},
⟨pad⟩     O = {100, },
⟨ugm⟩     O = {70,50},
⟨ppl|ptm⟩   \OE = {50,  },
⟨pad⟩   \OE = {100, },
⟨cmr⟩     P = {82,-50},
⟨pad|ppl|ptm⟩     P = {50,  },
⟨pmn⟩     P = {20,-50},
⟨bch|pmn|ppl|ptm⟩     Q = {50,  },
⟨cmr⟩     Q = {150,-30},
⟨pad⟩     Q = {100, },
⟨ugm⟩     Q = {70,50},
⟨cmr⟩     R = {75, 15},
⟨pad|ppl|ptm⟩     R = {50,  },
⟨pmn⟩     R = {20,  },
⟨bch|pad|ppl|ptm⟩     S = {50,  },
⟨cmr⟩     S = {90,-65},
⟨pmn⟩     S = {20,-30},
⟨bch|pad|ppl|ptm⟩     $ = {50,  },
⟨cmr⟩     $ = {100,-20},
⟨pmn⟩     $ = {20,-30},
⟨bch|pmn|ugm⟩     T = {70,  },
⟨cmr⟩      T = {220,-85},
⟨pad|ppl|ptm⟩      T = {100, },
⟨cmr⟩      U = {230,-55},
⟨pad|ppl|ptm⟩      U = {50,  },
⟨pmn⟩      U = {50,-50},
⟨cmr⟩      V = {260,-60},
⟨pad|pmn|ugm⟩      V = {100, },
⟨ppl|ptm⟩      V = {100,50},
⟨cmr⟩      W = {185,-55},
⟨pad|pmn|ugm⟩      W = {100, },
⟨ppl⟩      W = {50,  },
⟨ptm⟩      W = {100,50},
⟨cmr⟩      X = {70,-30},
⟨ppl|ptm⟩      X = {50,  },
⟨cmr⟩      Y = {250,-60},
⟨pmn⟩      Y = {50,  },
⟨ppl⟩      Y = {100,50},
⟨ptm⟩      Y = {100, },
⟨cmr⟩      Z = {90,-60},
⟨pmn⟩      Z = {  ,-50},
⟨cmr⟩      a = {150,-10},
⟨cmr⟩      b = {170,   },
⟨cmr⟩      c = {173,-10},
⟨cmr⟩      d = {150,-55},
⟨pmn⟩      d = {  ,-50},
⟨cmr⟩      e = {180, },
⟨cmr⟩      f = {  ,-250},
⟨pad|pmn⟩      f = { ,-100},
⟨cmr⟩      g = {150,-10},
⟨cmr⟩      h = {100, },
⟨cmr⟩      i = {210, },
⟨pmn⟩      i = {  ,-30},
⟨cmr⟩      j = {  ,-40},
⟨pmn⟩      j = {  ,-30},
⟨cmr⟩      k = {110,-50},
⟨cmr⟩      l = {240,-110},
⟨pmn⟩      l = {  ,-100},
⟨cmr⟩      m = {80, },
⟨cmr⟩      n = {115, },
⟨bch⟩      o = {50,50},
⟨cmr⟩      o = {155, },
⟨bch⟩      p = {  ,50},
⟨pmn⟩      p = {-50,  },
⟨bch⟩      q = {50,  },
⟨cmr⟩      q = {170,-40},
⟨cmr⟩      r = {155,-40},
⟨pmn⟩      r = {  ,50},
⟨cmr⟩      s = {130, },
⟨bch⟩      t = {  ,50},
⟨cmr⟩      t = {230,-10},
⟨cmr⟩      u = {120, },
⟨cmr⟩      v = {140,-25},
⟨pmn|ugm⟩      v = {50,  },
⟨bch⟩      w = {  ,50},
⟨cmr⟩      w = {98,-20},
⟨pmn|ugm⟩      w = {50,  },
⟨cmr⟩      x = {65,-40},
⟨bch⟩      y = {  ,50},
⟨cmr⟩      y = {130,-20},
⟨cmr⟩      z = {110,-80},
⟨cmr⟩      0 = {170,-85},
⟨bch|ptm⟩      1 = {150,100},
⟨cmr⟩      1 = {230,110},
⟨pad⟩      1 = {150, },
⟨pmn⟩      1 = {50,  },
⟨ppl⟩     1 = {100, },
⟨ugm⟩     1 = {150,150},
⟨cmr⟩     2 = {130,-70},
⟨pad|ppl|ptm⟩     2 = {50,  },
⟨pmn⟩     2 = {-50,  },
⟨bch⟩     3 = {50,  },
⟨cmr⟩     3 = {140,-70},
⟨pmn⟩     3 = {-100, },
⟨ptm⟩     3 = {100,50},
⟨bch⟩     4 = {100, },
⟨cmr⟩     4 = {130,80},
⟨pad⟩     4 = {150, },
⟨ppl|ptm⟩     4 = {50,  },
⟨cmr⟩     5 = {160, },
⟨ptm⟩     5 = {50,  },
⟨bch⟩     6 = {50,  },
⟨cmr⟩     6 = {175,-30},
⟨bch|pad|ptm⟩     7 = {100, },
⟨cmr⟩     7 = {250,-150},
⟨pmn⟩     7 = {20,  },
⟨ppl⟩     7 = {50,  },
⟨cmr⟩     8 = {130,-40},
⟨cmr⟩     9 = {155,-80},
⟨m-t|cmr|pad|pmn|ppl⟩     . = { ,500},
⟨blg⟩     . = {400,600},
⟨bch|ptm|ugm⟩     . = { ,700},
⟨blg⟩    {,}= {300,500},
⟨m-t|pad|pmn|ppl⟩    {,}= { ,500},
⟨cmr⟩    {,}= { ,450},
⟨bch|ugm⟩    {,}= { ,600},
⟨ptm⟩    {,}= { ,700},
⟨m-t|cmr|pad|ppl⟩     : = { ,300},
⟨bch|ugm⟩     : = { ,400},
⟨pmn⟩     : = { ,200},
⟨ptm⟩     : = { ,500},
⟨m-t|cmr|pad|ppl⟩     ; = { ,300},
⟨bch|ugm⟩     ; = { ,400},
⟨pmn⟩     ; = { ,200},
⟨ptm⟩     ; = { ,500},
⟨ptm⟩     ! = { ,100},
⟨bch⟩     ? = { ,200},
⟨ptm⟩     ? = { ,100},
⟨ppl⟩     ? = { ,300},
⟨pmn⟩     " = {400,200},
⟨m-t|pad|pmn|ppl|ptm⟩     & = {50,50},
⟨bch⟩     & = {  ,80},
⟨cmr⟩     & = {130,30},
⟨ugm⟩     & = {50,100},
⟨m-t|pad|pmn⟩    \% = {100, },
⟨cmr⟩    \% = {180,50},
⟨bch⟩    \% = {50,50},
⟨ppl|ptm⟩    \% = {100,100},
⟨ugm⟩    \% = {100,50},
⟨m-t|pmn|ppl⟩     * = {200,200},
⟨bch⟩     * = {300,200},
⟨cmr⟩     * = {380,20},
⟨pad⟩     * = {500,100},
⟨ptm|ugm⟩     * = {400,200},
⟨m-t|pmn|ppl⟩     + = {150,200},
⟨cmr⟩     + = {180,200},
⟨bch|ugm⟩     + = {250,250},
⟨pad|ptm⟩     + = {250,200},
⟨m-t|pad|pmn|ppl⟩     @ = {50,50},
⟨bch⟩     @ = {80,50},
⟨cmr⟩     @ = {180,10},
⟨ptm⟩     @ = {150,150},
⟨m-t|bch|ugm⟩     ~ = {150,150},
⟨cmr|pad|pmn|ppl|ptm⟩     ~ = {200,150},
⟨ugm⟩    {=}= {200,200},
⟨m-t|bch|pad|pmn|ppl|ptm|ugm⟩     ( = {200, },    ) = {  ,200},
⟨cmr⟩     ( = {300, },    ) = {  ,70},
⟨m-t|pad|ppl|ptm|ugm⟩     / = {100,200},
⟨cmr⟩     / = {100,100},
⟨bch⟩     / = {  ,150},
⟨pmn⟩     / = {100,150},
⟨m-t⟩     - = {300,300},
⟨bch|pad⟩     - = {300,400},
⟨pmn⟩     - = {200,300},
⟨cmr⟩     - = {500,300},
⟨ppl⟩     - = {300,500},
⟨ptm⟩     - = {500,500},
⟨ugm⟩     - = {400,700},
⟨blg⟩     _ = {0,300},
⟨m-t|pmn⟩     \textendash       = {200,200},   \textemdash        = {150,150},
⟨bch⟩     \textendash       = {200,300},   \textemdash        = {150,200},
⟨cmr⟩     \textendash       = {500,300},   \textemdash        = {400,170},
⟨pad|ppl|ptm|ugm⟩     \textendash       = {300,300},   \textemdash        = {200,200},
⟨m-t|bch|pmn|ugm⟩     \textquoteleft    = {400,200},   \textquoteright    = {400,200},
⟨blg⟩     \textquoteleft     = {400,400},    \textquoteright    = {400,400},
⟨cmr⟩     \textquoteleft    = {800,200},   \textquoteright    = {800,-20},
⟨pad⟩     \textquoteleft    = {800,200},   \textquoteright    = {800,200},
⟨ppl⟩     \textquoteleft    = {700,400},   \textquoteright    = {700,400},
⟨ptm⟩     \textquoteleft    = {800,500},   \textquoteright    = {800,500},
⟨m-t|bch|pmn⟩     \textquotedblleft = {400,200},   \textquotedblright = {400,200}
⟨blg⟩     \textquotedblright = {300,300}
⟨cmr⟩     \textquotedblleft = {540,100},   \textquotedblright = {500,100}
⟨pad⟩     \textquotedblleft = {700,200},   \textquotedblright = {700,200}
⟨ppl⟩     \textquotedblleft = {500,300},   \textquotedblright = {500,300}
⟨ptm⟩     \textquotedblleft = {700,400},   \textquotedblright = {700,400}
⟨ugm⟩     \textquotedblleft = {600,200},   \textquotedblright = {600,200}
   }

⟨*cmr|pmn⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-it-OT1,
⟨pmn⟩   [ name     = pmnj-it-OT1,
⟨cmr⟩     load     = cmr-it   ]
⟨pmn⟩     load     = pmnj-it  ]
⟨cmr⟩   { encoding = {OT1,OT4},
⟨pmn⟩   { encoding = OT1,
⟨cmr⟩     family   = cmr,
⟨pmn⟩     family   = pmnj,
⟨cmr⟩     shape    = it       }
⟨pmn⟩     shape    = {it,sl}  }
   {
⟨cmr⟩     \AE = {100,   },
⟨pmn⟩     \AE = {  ,-50},
⟨cmr⟩     \OE = {100,   },
⟨pmn⟩     \OE = {50,   }
⟨*cmr⟩
     "00 = {200,150}, % \Gamma
     "01 = {150,100}, % \Delta
     "02 = {150, 50}, % \Theta
     "03 = {150, 50}, % \Lambda
     "04 = {100,100}, % \Xi
     "05 = {100,100}, % \Pi
     "06 = {100, 50}, % \Sigma
     "07 = {200,150}, % \Upsilon
     "08 = {150, 50}, % \Phi
     "09 = {150,100}, % \Psi
     "OA = { 50, 50}  % \Omega
⟨/cmr⟩
   }

⟨/cmr|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = T1-it-default,
⟨bch⟩   [ name     = bch-it-T1,
⟨blg⟩   [ name     = blg-it-T1,
⟨cmr⟩   [ name     = cmr-it-T1,
⟨pad⟩   [ name     = pad-it-T1,
⟨pmn⟩   [ name     = pmnj-it-T1,
⟨ppl⟩   [ name     = ppl-it-T1,
⟨ptm⟩   [ name     = ptm-it-T1,
⟨ugm⟩   [ name     = ugm-it-T1,
⟨m-t⟩     load     = OT1-it   ]
⟨bch⟩     load     = bch-it   ]
⟨blg⟩     load     = blg-T1   ]
⟨cmr⟩     load     = cmr-it   ]
⟨pmn⟩     load     = pmnj-it  ]
⟨pad⟩     load     = pad-it   ]
⟨ppl⟩     load     = ppl-it   ]
⟨ptm⟩     load     = ptm-it   ]
⟨ugm⟩     load     = ugm-it   ]
⟨m-t|bch|cmr|pad|pmn|ppl⟩   { encoding = {T1,LY1},
⟨blg|ptm|ugm⟩   { encoding = T1,
⟨bch⟩     family   = bch,
⟨blg⟩     family   = blg,
⟨cmr⟩     family   = cmr,
⟨pmn⟩     family   = pmnj,
⟨pad⟩     family   = {pad,padx,padj},
⟨ppl⟩     family   = {ppl,pplx,pplj},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
⟨ugm⟩     family   = ugm,
⟨m-t|bch|pad|pmn|ppl|ptm⟩     shape    = {it,sl}  }
⟨blg|cmr|ugm⟩     shape    = it       }
   {
⟨m-t|bch|pmn⟩     _ = {  ,100},
⟨blg⟩     _ = {0,300},
⟨cmr|ugm⟩     _ = {100,200},
⟨pad|ppl|ptm⟩     _ = {100,100},
⟨blg⟩     . = {400,600},
⟨blg⟩    {,}= {300,500},
⟨cmr⟩     \AE = {100,   },
⟨pmn⟩     \AE = {  ,-50},
⟨bch|pmn⟩     \OE = { 50,   },
⟨cmr⟩     \OE = {100,   },
⟨pmn⟩     031 = { ,-100}, % ffl
⟨cmr|ptm⟩     156 = {100, },  % IJ
⟨pad⟩     156 = {50,  },  % IJ
⟨pmn⟩     156 = {20,  },  % IJ
⟨pmn⟩     188 = {  ,-30}, % ij
⟨pmn⟩   \v t = { ,100},
⟨m-t|pad|ppl|ptm⟩     \textbackslash    = {100,200},
⟨cmr|ugm⟩     \textbackslash    = {300,300},
⟨bch⟩     \textbackslash    = {150,150},
⟨pmn⟩     \textbackslash    = {100,150},
⟨ugm⟩     \textbar          = {200,200},
⟨cmr⟩     \textquotedblleft = {500,300},
⟨blg⟩    \textquoteleft     = {400,400},    \textquoteright    = {400,400},
⟨blg⟩    \textquotedbl      = {300,300},    \textquotedblleft  = {300,300},
⟨blg⟩    \textquotedblright = {300,300},    \quotedblbase      = {200,600},
⟨m-t|ptm⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {400,500},
⟨cmr⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {200,600},
⟨bch|pmn⟩     \quotesinglbase   = {200,500},   \quotedblbase      = {150,500},
⟨pad|ppl⟩     \quotesinglbase   = {500,500},   \quotedblbase      = {400,400},
⟨ugm⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {300,500},
⟨m-t|ppl|ptm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨bch|pmn⟩     \guilsinglleft    = {300,400},   \guilsinglright    = {200,500},
⟨cmr⟩     \guilsinglleft    = {500,300},   \guilsinglright    = {400,400},
⟨pad⟩     \guilsinglleft    = {500,400},   \guilsinglright    = {300,500},
⟨ugm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
⟨m-t|ppl⟩     \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨bch|pmn⟩     \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨pad⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ptm⟩     \guillemotleft    = {300,400},   \guillemotright    = {200,400},
⟨ugm⟩     \guillemotleft    = {300,400},   \guillemotright    = {300,400},
⟨m-t|pad|ppl|ugm⟩     \textexclamdown   = {100,   },   \textquestiondown  = {200,   },
⟨cmr|ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨pmn⟩     \textexclamdown   = {-50,   },   \textquestiondown  = {-50,   },
⟨m-t|ppl|ugm⟩     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
⟨bch|pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr|pad|ptm⟩     \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨bch|pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100},
⟨cmr|pad|ppl|ptm⟩     \textless         = {300,100},   \textgreater       = {200,100}
⟨pmn⟩     \textvisiblespace = {100,100}
  }

⟨*m-t|cmr|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = T2A-it-default,
⟨cmr⟩   [ name     = cmr-it-T2A,
⟨pmn⟩   [ name     = pmnj-it-T2A,
⟨m-t⟩     load     = OT1-it   ]
⟨cmr⟩     load     = cmr-it   ]
⟨pmn⟩     load     = pmnj-it  ]
  { encoding = T2A,
⟨cmr⟩     family   = cmr,
⟨pmn⟩     family   = pmnj,
⟨m-t|pmn⟩     shape    = {it,sl}  }
⟨cmr⟩     shape    = it       }
  {
⟨cmr⟩     \CYRA = {100,50},
⟨pmn⟩     \CYRA = {50,  },
⟨cmr⟩     \CYRB = {50,  },
⟨cmr⟩     \CYRV = {50,  },
⟨pmn⟩     \CYRV = {20,-50},
⟨cmr⟩     \CYRG = {100, },
⟨pmn⟩     \CYRG = {10,  },
⟨cmr⟩     \CYRD = {50,  },
⟨cmr⟩     \CYRE = {50,  },
⟨pmn⟩     \CYRE = {20,-50},
⟨cmr⟩     \CYRZH = {50,  },
⟨cmr⟩     \CYRZ = {50,  },
⟨pmn⟩     \CYRZ = {20,-50},
⟨cmr⟩     \CYRI = {50,  },
⟨pmn⟩     \CYRI = {  ,-30},
⟨cmr⟩     \CYRISHRT = {50,  },
⟨cmr⟩     \CYRK = {50,  },
⟨pmn⟩     \CYRK = {20,  },
⟨cmr⟩     \CYRL = {50,  },
⟨cmr⟩     \CYRM = {50,  },
⟨pmn⟩     \CYRM = {  ,-30},
⟨cmr⟩     \CYRN = {50,  },
⟨cmr⟩     \CYRO = {100, },
⟨pmn⟩     \CYRO = {50,  },
⟨cmr⟩     \CYRP = {50,  },
⟨cmr⟩     \CYRR = {50,  },
⟨pmn⟩     \CYRR = {20,-50},
⟨cmr⟩     \CYRS = {100, },
⟨pmn⟩     \CYRS = {50,  },
⟨cmr⟩     \CYRT = {100, },
⟨pmn⟩     \CYRT = {70,  },
⟨cmr⟩     \CYRU = {100, },
⟨pmn⟩     \CYRU = {50,  },
⟨cmr⟩     \CYRF = {100, },
⟨cmr⟩     \CYRH = {50,  },
⟨cmr⟩     \CYRC = {50,  },
⟨cmr⟩     \CYRCH = {100, },
⟨cmr⟩     \CYRSH = {50,  },
⟨cmr⟩     \CYRSHCH = {50,  },
⟨cmr⟩     \CYRHRDSN = {100, },
⟨cmr⟩     \CYRERY = {50,  },
⟨cmr⟩     \CYRSFTSN = {50,  },
⟨cmr⟩     \CYREREV = {50,  },
⟨cmr⟩     \CYRYU = {50,  },
⟨cmr⟩     \CYRYA = {50,  },
⟨pmn⟩     \CYRYA = {  ,20},
⟨pmn⟩     \cyrr = {-50,  },
⟨m-t|pmn⟩      _ = {  ,100},
⟨cmr⟩      _ = {100,200},
⟨pmn⟩      031 = { ,-100}, % ffl
⟨pmn⟩     \v t = { ,100},
⟨m-t⟩     \textbackslash    = {100,200},   \quotedblbase      = {400,500},
⟨cmr⟩     \textbackslash    = {300,300},   \quotedblbase      = {200,600},
⟨pmn⟩     \textbackslash    = {100,150},   \quotedblbase      = {150,500},
⟨m-t⟩     \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨pmn⟩     \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨m-t⟩     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
⟨cmr⟩     \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr⟩     \textquotedblleft = {500,300},
⟨cmr⟩     \textless         = {300,100},   \textgreater       = {200,100}
⟨pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100}
  }

⟨/m-t|cmr|pmn⟩
⟨*m-t|ptm⟩
\SetProtrusion
⟨m-t⟩   [ name     = QX-it-default,
⟨ptm⟩   [ name     = ptm-it-QX,
⟨m-t⟩     load     = OT1-it ]
⟨ptm⟩     load     = ptm-it ]
   { encoding = {QX},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
     shape    = {it,sl}  }
   {
⟨ptm⟩     009 = {  , 50}, % fk
     {=} = {100,100},
⟨m-t⟩     \textunderscore   = {100,100},
⟨ptm⟩     \textunderscore   = {100,150},
     \textbackslash    = {100,200},
     \quotedblbase     = {300,400},
⟨m-t⟩     \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨ptm⟩     \guillemotleft    = {200,400},   \guillemotright    = {200,400},
     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
     \textless         = {100,100},   \textgreater       = {100,100},
     \textminus        = {200,200},   \textdegree        = {300,150},
⟨m-t⟩     \copyright        = {100,100},   \textregistered    = {100,100}
⟨ptm⟩     \textregistered   = {100,150},   \copyright         = {100,150},
⟨ptm⟩     \textDelta        = { 70,   },   \textdelta         = {   , 50},
⟨ptm⟩     \textpi           = { 50, 80},   \textmu            = {   , 80},
⟨ptm⟩     \texteuro         = {200,   },  \textellipsis      = {100,200},
⟨ptm⟩     \textquoteleft    = {500,400},  \textquoteright    = {500,400},
⟨ptm⟩     \textquotedblleft = {500,300},  \textquotedblright = {400,400},
⟨ptm⟩     \textapprox       = { 50, 50},  \textinfty         = {100,100},
⟨ptm⟩     \textdagger       = {150,150},  \textdaggerdbl     = {100,100},
⟨ptm⟩     \textdiv          = {150,150},  \textasciitilde    = { 80, 80},
⟨ptm⟩     \texttimes        = {100,150},  \textpm            = { 50, 80},
⟨ptm⟩     \textbullet       = {300,100},  \textperiodcentered = {300,300},
⟨ptm⟩     \textquotesingle  = {500,500},  \textquotedbl      = {300,300},
⟨ptm⟩     \textperthousand  = {    ,50}
   }

⟨/m-t|ptm⟩
⟨*cmr|bch⟩
\SetProtrusion
⟨cmr⟩   [ name = cmr-it-T5,
⟨cmr⟩     load = cmr-it ]
⟨bch⟩   [ name = bch-it-T5,
⟨bch⟩     load = bch-it ]
   { encoding = T5,
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
     shape    = it }
   {
⟨bch⟩      _ = {  ,100},
⟨cmr⟩      _ = {100,200},
⟨bch⟩     \textbackslash    = {150,150},
⟨cmr⟩     \textbackslash    = {300,300},
⟨bch⟩     \quotesinglbase   = {200,500},  \quotedblbase      = {150,500},
⟨cmr⟩     \quotesinglbase   = {300,700},  \quotedblbase      = {200,600},
⟨bch⟩     \guilsinglleft    = {300,400},  \guilsinglright    = {200,500},
⟨cmr⟩     \guilsinglleft    = {500,300},  \guilsinglright    = {400,400},
⟨bch⟩     \guillemotleft    = {200,300},  \guillemotright    = {150,400},
⟨cmr⟩     \guillemotleft    = {400,100},  \guillemotright    = {200,300},
⟨bch⟩     \textbraceleft    = {200,   },  \textbraceright    = {   ,200},
⟨cmr⟩     \textbraceleft    = {400,100},  \textbraceright    = {200,200},
⟨bch⟩     \textless         = {100,   },  \textgreater       = {   ,100}
⟨cmr⟩     \textless         = {300,100},  \textgreater       = {200,100}
  }

⟨/cmr|bch⟩
```

Slanted is very similar to italic.

```
⟨*cmr⟩
\SetProtrusion
   [ name     = cmr-sl,
     load     = cmr-it-OT1 ]
   { encoding = {OT1,OT4},
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T1,
     load     = cmr-it-T1 ]
   { encoding = {T1,LY1},
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T2A,
     load     = cmr-it-T2A ]
   { encoding = T2A,
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T5,
     load     = cmr-it-T5 ]
   { encoding = T5,
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = lmr-it-T1,
     load     = cmr-it-T1 ]
   { encoding = {T1,LY1},
     family   = lmr,
     shape    = {it,sl} }
   {
     \textquotedblleft  = { ,200},  \textquotedblright = { ,200},
     \quotesinglbase    = { ,400},  \quotedblbase      = { ,500}
   }

```

Oldstyle numerals are slightly different.

```
\SetProtrusion
   [ name = cmr(oldstyle)-it,
     load = cmr-it-T1 ]
   { encoding = T1,
     family   = {hfor,cmor},
     shape    = {it,sl}  }
   {
     1 = {250, 50},
     2 = {150,-100},
     3 = {100,-50},
     4 = {150,150},
     6 = {200,   },
     7 = {200, 50},
     8 = {150,-50},
     9 = {100, 50}
   }

⟨/cmr⟩
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-it,
     load     = pmnj-it ]
   { encoding = OT1,
     family   = pmnx,
     shape    = {it,sl} }
   {
     1 = {100,150}
   }

\SetProtrusion
   [ name     = pmnx-it-T1,
     load     = pmnj-it-T1 ]
   { encoding = {T1,LY1},
     family   = pmnx,
     shape    = {it,sl} }
   {
     1 = {100,150}
   }

\SetProtrusion
   [ name     = pmnx-it-T2A,
     load     = pmnj-it-T2A ]
   { encoding = {T2A},
     family   = pmnx,
     shape    = {it,sl} }
   {
     1 = {100,150}
   }

⟨/pmn⟩
⟨*ptm⟩
\SetProtrusion
   [ name     = ptm-it-LY1,
     load     = ptm-it-T1   ]
   { encoding = {LY1},
     family   = {ptm,ptmx,ptmj},
     shape    = {it,sl}  }
   {
     _                        = {100,100},
     \texttrademark           = {100,100},
     \textregistered          = {100,100},
     \textcopyright           = {100,100},
     \textdegree              = {300,100},
     \textminus               = {200,200},
     \textellipsis            = {100,200},
%    \texteuro                = {   ,   }, % ?
     \textcent                = {100,100},
     \textquotesingle         = {500,   },
     \textflorin              = {100, 70},
     \textdagger              = {150,150},
     \textdaggerdbl           = {100,100},
     \textbullet              = {150,150},
     \textonesuperior         = {150,100},
     \texttwosuperior         = {150, 50},
     \textthreesuperior       = {150, 50},
     \textparagraph           = {100,   },
     \textperiodcentered      = {500,300},
     \textonequarter          = { 50,   },
     \textonehalf             = { 50,   },
     \textplusminus           = {100,100},
     \textmultiply            = {150,150},
     \textdivide              = {150,150}
   }

⟨/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).

```
⟨*!(blg|ugm)⟩
\SetProtrusion
⟨m-t⟩   [ name     = OT1-sc,
⟨bch⟩   [ name     = bch-sc,
⟨cmr⟩   [ name     = cmr-sc-OT1,
⟨pad⟩   [ name     = pad-sc,
⟨pmn⟩   [ name     = pmnj-sc,
⟨ppl⟩   [ name     = ppl-sc,
⟨ptm⟩   [ name     = ptm-sc,
⟨m-t⟩     load     = default ]
⟨bch⟩     load     = bch-default ]
⟨cmr⟩     load     = cmr-OT1 ]
⟨pad⟩     load     = pad-default ]
⟨pmn⟩     load     = pmnj-default ]
⟨ppl⟩     load     = ppl-default ]
⟨ptm⟩     load     = ptm-default ]
⟨m-t|bch|pad|pmn⟩   { encoding = OT1,
⟨cmr|ppl|ptm⟩   { encoding = {OT1,OT4},
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
⟨pad⟩     family   = {pad,padx,padj},
⟨pmn⟩     family   = pmnj,
⟨ppl⟩     family   = {ppl,pplx,pplj},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
     shape    = sc }
   {
     a = {50,50},
⟨cmr|pad|ppl|ptm⟩   \ae = {50,  },
⟨bch|pmn⟩     c = {50,  },
⟨bch|pad|pmn⟩     d = {  ,50},
⟨m-t|bch|cmr|pad|pmn|ptm⟩     f = {  ,50},
⟨bch|pad|pmn⟩     g = {50,  },
⟨m-t|cmr|pad|pmn|ppl|ptm⟩     j = {50,  },
⟨bch⟩     j = {100,  },
⟨m-t|bch|cmr|pad|pmn|ppl⟩     l = {  ,50},
⟨ptm⟩     l = {  ,80},
⟨m-t|bch|cmr|pad|pmn|ppl⟩   013 = {  ,50}, % fl
⟨ptm⟩   013 = {  ,80}, % fl
⟨bch|pad|pmn⟩     o = {50,50},
⟨pad|pmn⟩   \oe = {50,  },
⟨ppl⟩     p = { 0, 0},
⟨bch|pad|pmn⟩     q = {50,70},
⟨ppl⟩     q = { 0,  },
⟨m-t|cmr|pad|pmn|ppl|ptm⟩     r = {  , 0},
     t = {50,50},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     y = {50,50}
⟨ptm⟩     y = {80,80}
   }

\SetProtrusion
⟨m-t⟩   [ name     = T1-sc,
⟨bch⟩   [ name     = bch-sc-T1,
⟨cmr⟩   [ name     = cmr-sc-T1,
⟨pad⟩   [ name     = pad-sc-T1,
⟨pmn⟩   [ name     = pmnj-sc-T1,
⟨ppl⟩   [ name     = ppl-sc-T1,
⟨ptm⟩   [ name     = ptm-sc-T1,
⟨m-t⟩     load     = T1-default ]
⟨bch⟩     load     = bch-T1     ]
⟨cmr⟩    load      = cmr-T1     ]
⟨pad⟩    load      = pad-T1     ]
⟨pmn⟩    load      = pmnj-T1    ]
⟨ppl⟩    load      = ppl-T1     ]
⟨ptm⟩    load      = ptm-T1     ]
   { encoding = {T1,LY1},
⟨bch⟩    family   = bch,
⟨cmr⟩    family   = cmr,
⟨pad⟩    family   = {pad,padx,padj},
⟨pmn⟩    family   = pmnj,
⟨ppl⟩    family   = {ppl,pplx,pplj},
⟨ptm⟩    family   = {ptm,ptmx,ptmj},
     shape    = sc }
   {
     a = {50,50},
⟨cmr|pad|ppl|ptm⟩ \ae = {50,  },
⟨bch|pmn⟩     c = {50,  },
⟨bch|pad|pmn⟩     d = {  ,50},
⟨m-t|bch|cmr|pad|pmn|ptm⟩     f = {  ,50},
⟨bch|pad|pmn⟩     g = {50,  },
⟨m-t|cmr|pad|pmn|ppl|ptm⟩     j = {50,  },
⟨bch⟩     j = {100,  },
⟨m-t|bch|cmr|pad|pmn|ppl⟩     l = {  ,50},
⟨ptm⟩     l = {  ,80},
⟨m-t|bch|cmr|pad|pmn|ppl⟩   029 = {  ,50}, % fl
⟨ptm⟩   029 = {  ,80}, % fl
⟨bch|pad|pmn⟩     o = {50,50},
⟨bch|pad|pmn⟩   \oe = {50,  },
⟨ppl⟩     p = { 0, 0},
⟨bch|pad|pmn⟩     q = {50,70},
⟨ppl⟩     q = { 0,  },
⟨m-t|cmr|pad|pmn|ppl|ptm⟩     r = {  , 0},
     t = {50,50},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     y = {50,50}
⟨ptm⟩     y = {80,80}
   }

⟨/!(blg|ugm)⟩
⟨*m-t|cmr⟩
\SetProtrusion
⟨m-t⟩   [ name      = T2A-sc,
⟨cmr⟩   [ name      = cmr-sc-T2A,
⟨m-t⟩     load      = T2A-default ]
⟨cmr⟩     load      = cmr-T2A     ]
   { encoding = T2A,
⟨cmr⟩     family   = cmr,
     shape    = sc }
   {
     \cyra = {50,50},
     \cyrg = {  ,50},
     \cyrt = {50,50},
     \cyry = {  ,50}
   }

⟨/m-t|cmr⟩
⟨*m-t⟩
\SetProtrusion
   [ name      = QX-sc,
     load      = QX-default ]
   { encoding = QX,
     shape    = sc  }
   {
     a = {50,50},
     f = {  ,50},
     j = {50,  },
    l = {  ,50},
  013 = {  ,50}, % fl
    r = {  , 0},
    t = {50,50},
    y = {50,50}
  }

⟨/m-t⟩
⟨*cmr|bch⟩
\SetProtrusion
⟨bch⟩   [ name     = bch-sc-T5,
⟨bch⟩     load     = bch-T5 ]
⟨cmr⟩   [ name     = cmr-sc-T5,
⟨cmr⟩     load     = cmr-T5 ]
  { encoding = T5,
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
    shape    = sc }
  {
    a = {50,50},
⟨bch⟩     c = {50,  },
⟨bch⟩     d = {  ,50},
    f = {  ,50},
⟨bch⟩     g = {50,  },
⟨bch⟩     j = {100, },
⟨cmr⟩     j = {50,  },
    l = {  ,50},
⟨bch⟩     o = {50,50},
⟨bch⟩     q = { 0,  },
⟨cmr⟩     r = {  , 0},
    t = {50,50},
    y = {50,50}
  }

⟨/cmr|bch⟩
⟨*pmn⟩
\SetProtrusion
  [ name     = pmnx-sc,
    load     = pmnj-sc ]
  { encoding = OT1,
    family   = pmnx,
    shape    = sc }
  {
    1 = {230,180}
  }

\SetProtrusion
  [ name     = pmnx-sc-T1,
    load     = pmnj-sc-T1 ]
  { encoding = {T1,LY1},
    family   = pmnx,
    shape    = sc }
  {
    1 = {230,180}
  }

```

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`.

```
\SetProtrusion
  [ name     = pmnj-scit,
    load     = pmnj-it   ]
```

```
   { encoding = OT1,
     family   = pmnj,
     shape    = {scit,si} }
   {
     a = {50,  },
   \ae = {  ,-50},
     b = {20,-50},
     c = {50,-50},
     d = {20, 0},
     e = {20,-50},
     f = {10, 0},
   012 = {10,-50}, % fi
   013 = {10,-50}, % fl
   014 = {10,-50}, % ffi
   015 = {10,-50}, % ffl
     g = {50,-50},
     i = {20,-50},
     j = {20, 0},
     k = {20,  },
     l = {20,50},
     m = {  ,-30},
     n = {  ,-30},
     o = {50,  },
   \oe = {50,-50},
     p = {20,-50},
     q = {50,  },
     r = {20, 0},
     s = {20,-30},
     t = {70,  },
     u = {50,-50},
     v = {100,  },
     w = {100,  },
     y = {50,  },
     z = {  ,-50}
   }

\SetProtrusion
   [ name     = pmnj-scit-T1,
     load     = pmnj-it-T1   ]
   { encoding = {T1,LY1},
     family   = pmnj,
     shape    = {scit,si}    }
   {
     a = {50,  },
   \ae = {  ,-50},
     b = {20,-50},
     c = {50,-50},
     d = {20, 0},
     e = {20,-50},
     f = {10, 0},
   028 = {10,-50}, % fi
   029 = {10,-50}, % fl
   030 = {10,-50}, % ffi
   031 = {10,-50}, % ffl
     g = {50,-50},
     i = {20,-50},
   188 = {20, 0},  % ij
     j = {20, 0},
     k = {20,  },
     l = {20,50},
     m = {  ,-30},
     n = {  ,-30},
     o = {50,  },
   \oe = {50,-50},
     p = {20,-50},
     q = {50,  },
     r = {20, 0},
     s = {20,-30},
     t = {70,  },
     u = {50,-50},
     v = {100,  },
     w = {100,  },
     y = {50,  },
     z = {  ,-50}
   }

\SetProtrusion
   [ name     = pmnx-scit,
     load     = pmnj-scit ]
   { encoding = OT1,
     family   = pmnx,
     shape    = {scit,si} }
   {
     1 = {100,150}
   }

\SetProtrusion
   [ name     = pmnx-scit-T1,
     load     = pmnj-scit-T1 ]
   { encoding = {T1,LY1},
     family   = pmnx,
     shape    = {scit,si}    }
   {
     1 = {100,150}
   }

⟨/pmn⟩
```

15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```
\SetProtrusion
⟨m-t⟩   [ name     = textcomp ]
⟨bch⟩   [ name     = bch-textcomp ]
⟨blg⟩   [ name     = blg-textcomp ]
⟨cmr⟩   [ name     = cmr-textcomp ]
⟨pad⟩   [ name     = pad-textcomp ]
⟨pmn⟩   [ name     = pmn-textcomp ]
⟨ppl⟩   [ name     = ppl-textcomp ]
⟨ptm⟩   [ name     = ptm-textcomp ]
⟨ugm⟩   [ name     = ugm-textcomp ]
⟨m-t⟩   { encoding = TS1      }
⟨!m-t⟩   { encoding = TS1,
⟨bch⟩     family   = bch }
⟨blg⟩     family   = blg }
⟨cmr⟩     family   = cmr }
⟨pad⟩     family   = {pad,padx,padj} }
⟨pmn⟩     family   = {pmnx,pmnj} }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
⟨ugm⟩     family   = ugm }
   {
⟨blg⟩     \textquotestraightbase    = {400,500},
⟨cmr⟩     \textquotestraightbase    = {300,300},
⟨pad|pmn⟩     \textquotestraightbase    = {400,400},
⟨blg⟩     \textquotestraightdblbase = {300,400},
⟨cmr|pmn⟩     \textquotestraightdblbase = {300,300},
⟨pad⟩     \textquotestraightdblbase = {400,400},
⟨bch|cmr|pad|pmn|ugm⟩     \texttwelveudash          = {200,200},
⟨bch|cmr|pad|pmn⟩     \textthreequartersemdash  = {150,150},
⟨ugm⟩     \textthreequartersemdash  = {200,200},
⟨blg⟩     \textquotesingle          = {500,600},
⟨cmr|pmn⟩     \textquotesingle          = {300,400},
⟨pad⟩     \textquotesingle          = {400,500},
⟨ptm⟩     \textquotesingle          = {500,500},
⟨ugm⟩     \textquotesingle          = {300,500},
⟨bch|cmr|pmn⟩     \textasteriskcentered     = {200,300},
⟨blg⟩     \textasteriskcentered     = {150,200},
⟨pad⟩     \textasteriskcentered     = {300,300},
⟨ugm⟩     \textasteriskcentered     = {100,200},
⟨pmn⟩     \textfractionsolidus      = {-200,-200},
⟨cmr⟩     \textoneoldstyle          = {100,100},
⟨pmn⟩     \textoneoldstyle          = {   , 50},
⟨cmr⟩     \textthreeoldstyle        = {   , 50},
⟨pad|pmn⟩     \textthreeoldstyle        = { 50,   },
⟨cmr⟩     \textfouroldstyle         = { 50, 50},
⟨pad|pmn⟩     \textfouroldstyle         = { 50,   },
⟨cmr|pad|pmn⟩     \textsevenoldstyle        = { 50, 80},
⟨cmr⟩     \textlangle               = {400,   },
⟨cmr⟩     \textrangle               = {   ,400},
⟨m-t|bch|pmn|ptm⟩     \textminus                = {200,200},
⟨cmr|pad|ppl⟩     \textminus                = {300,300},
⟨blg|ugm⟩     \textminus                = {250,300},
⟨bch|pad|pmn⟩     \textlbrackdbl            = {100,   },
⟨blg⟩     \textlbrackdbl            = {200,   },
⟨bch|pad|pmn⟩     \textrbrackdbl            = {   ,100},
⟨blg⟩     \textrbrackdbl            = {   ,200},
⟨pmn⟩     \textasciigrave           = {200,500},
⟨bch|blg|cmr|pad|pmn⟩     \texttildelow             = {200,250},
⟨pmn⟩     \textasciibreve           = {300,400},
⟨pmn⟩     \textasciicaron           = {300,400},
⟨pmn⟩     \textacutedbl             = {200,300},
⟨pmn⟩     \textgravedbl             = {150,300},
⟨bch|pmn|ugm⟩     \textdagger               = { 80, 80},
⟨blg⟩     \textdagger               = {200,200},
⟨cmr|pad⟩     \textdagger               = {100,100},
⟨ptm⟩     \textdagger               = {150,150},
⟨blg⟩     \textdaggerdbl            = {150,150},
⟨cmr|pad|pmn⟩     \textdaggerdbl            = { 80, 80},
⟨ptm⟩     \textdaggerdbl            = {100,100},
⟨bch⟩     \textbardbl               = {100,100},
⟨blg|ugm⟩     \textbardbl               = {150,150},
⟨bch⟩     \textbullet               = {200,200},
⟨blg⟩     \textbullet               = {400,500},
⟨cmr|pad|pmn⟩     \textbullet               = {   ,100},
⟨ptm⟩     \textbullet               = {150,150},
⟨ugm⟩     \textbullet               = { 50,100},
⟨bch|cmr|pmn⟩     \textcelsius              = { 50,   },
⟨pad⟩     \textcelsius              = { 80,   },
⟨bch⟩     \textflorin               = { 50, 50},
⟨blg⟩     \textflorin               = {100,100},
⟨pad|ugm⟩     \textflorin               = {   ,100},
⟨pmn⟩     \textflorin               = { 50,100},
⟨ptm⟩     \textflorin               = { 50, 70},
⟨cmr⟩     \textcolonmonetary        = {   , 50},
⟨pad|pmn⟩     \textcolonmonetary        = { 50,   },
⟨pmn⟩     \textinterrobang          = {   ,100},
⟨pmn⟩     \textinterrobangdown      = {100,   },
⟨m-t|pad|ptm⟩     \texttrademark            = {100,100},
⟨bch⟩     \texttrademark            = {150,150},
⟨blg|cmr|ppl⟩     \texttrademark            = {200,200},
⟨pmn⟩     \texttrademark            = { 50, 50},
⟨ugm⟩     \texttrademark            = {100,150},
⟨bch|ugm⟩      \textcent                 = { 50,   },
⟨ptm⟩      \textcent                 = {100,100},
⟨bch⟩      \textsterling             = { 50,   },
⟨ugm⟩      \textsterling             = {   , 50},
⟨bch⟩      \textbrokenbar            = {200,200},
⟨blg⟩      \textbrokenbar            = {250,250},
⟨ugm⟩      \textbrokenbar            = {200,300},
⟨pmn⟩      \textasciidieresis        = {300,400},
⟨m-t|bch|cmr|pad|ptm|ugm⟩      \textcopyright            = {100,100},
⟨pmn⟩      \textcopyright            = {100,150},
⟨ppl⟩      \textcopyright            = {200,200},
⟨bch|cmr|ugm⟩      \textordfeminine          = {100,200},
⟨pad|pmn⟩      \textordfeminine          = {200,200},
⟨bch|cmr|pad|pmn|ugm⟩      \textlnot                 = {200,   },
⟨blg⟩      \textlnot                 = {200,100},
⟨m-t|bch|cmr|pad|ptm|ugm⟩      \textregistered           = {100,100},
⟨pmn⟩      \textregistered           = { 50,150},
⟨ppl⟩      \textregistered           = {200,200},
⟨pmn⟩      \textasciimacron          = {150,200},
⟨m-t|ppl|ptm⟩      \textdegree               = {300,300},
⟨bch⟩      \textdegree               = {150,200},
⟨blg|ugm⟩      \textdegree               = {200,200},
⟨cmr|pad⟩      \textdegree               = {400,400},
⟨pmn⟩      \textdegree               = {150,400},
⟨bch|cmr|pad|pmn|ugm⟩      \textpm                   = {150,200},
⟨blg⟩      \textpm                   = {100,100},
⟨ptm⟩      \textpm                   = { 50, 80},
⟨bch|blg|ugm⟩      \texttwosuperior          = {100,200},
⟨cmr⟩      \texttwosuperior          = { 50,100},
⟨pad|pmn⟩      \texttwosuperior          = {200,200},
⟨ptm⟩      \texttwosuperior          = { 50, 50},
⟨bch|blg|ugm⟩      \textthreesuperior        = {100,200},
⟨cmr⟩      \textthreesuperior        = { 50,100},
⟨pad|pmn⟩      \textthreesuperior        = {200,200},
⟨ptm⟩      \textthreesuperior        = { 50, 50},
⟨pmn⟩      \textasciiacute           = {300,400},
⟨bch|ugm⟩      \textmu                   = {   ,100},
⟨bch|pad|pmn⟩      \textparagraph            = {   ,100},
⟨bch|cmr|pad|pmn⟩      \textperiodcentered       = {300,400},
⟨blg⟩      \textperiodcentered       = {400,500},
⟨ptm⟩      \textperiodcentered       = {300,300},
⟨ugm⟩      \textperiodcentered       = {200,500},
⟨bch|blg|ugm⟩      \textonesuperior          = {200,300},
⟨cmr|pad|pmn⟩      \textonesuperior          = {200,200},
⟨ptm⟩      \textonesuperior          = {100,100},
⟨bch|pad|pmn|ugm⟩      \textordmasculine         = {200,200},
⟨blg|cmr⟩      \textordmasculine         = {100,200},
⟨bch|cmr|pmn⟩      \texteuro                 = {100,   },
⟨pad⟩      \texteuro                 = { 50,100},
⟨bch⟩      \texttimes                = {200,200},
⟨blg|ptm⟩      \texttimes                = {100,100},
⟨cmr⟩      \texttimes                = {150,250},
⟨pad⟩      \texttimes                = {100,150},
⟨pmn⟩      \texttimes                = { 70,100},
⟨ugm⟩      \texttimes                = {200,300},
⟨bch|pad|pmn⟩      \textdiv                  = {150,200}
⟨blg⟩      \textdiv                  = {100,100}
⟨cmr⟩      \textdiv                  = {150,250}
⟨ptm⟩      \textdiv                  = { 50,100},
⟨ugm⟩      \textdiv                  = {200,300},
⟨ptm⟩      \textperthousand          = {    ,50}
⟨ugm⟩      \textsection              = {   ,100},
⟨ugm⟩      \textonehalf              = { 50,100},
⟨ugm⟩      \textonequarter           = { 50,100},
⟨ugm⟩      \textthreequarters        = { 50,100},
⟨ugm⟩     \textsurd                 = {   ,100}
```

Remaining slots in the source file.

```
   }

⟨*cmr|pad|pmn|ugm⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-textcomp-it ]
⟨pad⟩   [ name     = pad-textcomp-it ]
⟨pmn⟩   [ name     = pmn-textcomp-it ]
⟨ugm⟩   [ name     = ugm-textcomp-it ]
   { encoding = TS1,
⟨cmr⟩     family   = cmr,
⟨pad⟩     family   = {pad,padx,padj},
⟨pmn⟩     family   = {pmnx,pmnj},
⟨ugm⟩     family   = ugm,
⟨!ugm⟩     shape    = {it,sl} }
⟨ugm⟩     shape    = it }
   {
⟨cmr⟩     \textquotestraightbase    = {300,600},
⟨pad|pmn⟩     \textquotestraightbase    = {400,400},
⟨cmr⟩     \textquotestraightdblbase = {300,600},
⟨pad⟩     \textquotestraightdblbase = {300,400},
⟨pmn⟩     \textquotestraightdblbase = {300,300},
     \texttwelveudash          = {200,200},
⟨cmr|pad|pmn⟩     \textthreequartersemdash  = {150,150},
⟨ugm⟩     \textthreequartersemdash  = {200,200},
⟨cmr⟩     \textquotesingle          = {600,300},
⟨pad⟩     \textquotesingle          = {800,100},
⟨pmn⟩     \textquotesingle          = {300,200},
⟨ugm⟩     \textquotesingle          = {500,500},
⟨cmr⟩     \textasteriskcentered     = {300,200},
⟨pad⟩     \textasteriskcentered     = {500,100},
⟨pmn⟩     \textasteriskcentered     = {200,300},
⟨ugm⟩     \textasteriskcentered     = {300,150},
⟨pmn⟩     \textfractionsolidus      = {-200,-200},
⟨cmr⟩     \textoneoldstyle          = {100, 50},
⟨pad⟩     \textoneoldstyle          = {100,   },
⟨pmn⟩     \textoneoldstyle          = { 50,   },
⟨pad⟩     \texttwooldstyle          = { 50,   },
⟨pmn⟩     \texttwooldstyle          = {-50,   },
⟨cmr⟩     \textthreeoldstyle        = {100, 50},
⟨pmn⟩     \textthreeoldstyle        = {-100,  },
⟨cmr⟩     \textfouroldstyle         = { 50, 50},
⟨pad⟩     \textfouroldstyle         = { 50,100},
⟨cmr⟩     \textsevenoldstyle        = { 50, 80},
⟨pad⟩     \textsevenoldstyle        = { 50,   },
⟨pmn⟩     \textsevenoldstyle        = { 20,   },
⟨cmr⟩     \textlangle               = {400,   },
⟨cmr⟩     \textrangle               = {   ,400},
⟨cmr|pad⟩     \textminus                = {300,300},
⟨pmn⟩     \textminus                = {200,200},
⟨ugm⟩     \textminus                = {250,300},
⟨pad|pmn⟩     \textlbrackdbl            = {100,   },
⟨pad|pmn⟩     \textrbrackdbl            = {   ,100},
⟨pmn⟩     \textasciigrave           = {300,300},
⟨cmr|pad|pmn⟩     \texttildelow             = {200,250},
⟨pmn⟩     \textasciibreve           = {300,300},
⟨pmn⟩     \textasciicaron           = {300,300},
⟨pmn⟩     \textacutedbl             = {200,300},
⟨pmn⟩     \textgravedbl             = {150,300},
⟨cmr⟩     \textdagger               = {100,100},
⟨pad⟩     \textdagger               = {200,100},
⟨pmn⟩     \textdagger               = { 80, 50},
⟨ugm⟩     \textdagger               = { 80, 80},
⟨cmr|pad⟩     \textdaggerdbl            = { 80, 80},
⟨pmn⟩     \textdaggerdbl            = { 80, 50},
⟨ugm⟩     \textbardbl               = {150,150},
⟨cmr⟩     \textbullet               = {200,100},
⟨pad⟩     \textbullet               = {300,   },
⟨pmn⟩     \textbullet               = { 30, 70},
⟨ugm⟩     \textbullet               = { 50,100},
⟨cmr⟩     \textcelsius              = {100,   },
⟨pad⟩     \textcelsius              = {200,   },
⟨pmn⟩     \textcelsius              = { 50,-50},
⟨pad⟩     \textflorin               = {100,   },
⟨pmn⟩     \textflorin               = { 50,100},
⟨ugm⟩     \textflorin               = {   ,100},
⟨cmr⟩     \textcolonmonetary        = {150,   },
⟨pad⟩     \textcolonmonetary        = {100,   },
⟨pmn⟩     \textcolonmonetary        = { 50,-50},
⟨cmr|pad⟩     \texttrademark            = {200,   },
⟨pmn⟩     \texttrademark            = { 50,100},
⟨ugm⟩     \texttrademark            = {150, 50},
⟨ugm⟩     \textcent                 = { 50,   },
⟨ugm⟩     \textsterling             = {   , 50},
⟨ugm⟩     \textbrokenbar            = {200,300},
⟨pmn⟩     \textasciidieresis        = {300,200},
⟨cmr⟩     \textcopyright            = {100,   },
⟨pad⟩     \textcopyright            = {200,100},
⟨pmn⟩     \textcopyright            = {100,150},
⟨ugm⟩     \textcopyright            = {300,   },
⟨cmr⟩     \textordfeminine          = {100,100},
⟨pmn⟩     \textordfeminine          = {200,200},
⟨ugm⟩     \textordfeminine          = {100,200},
⟨cmr|pad⟩     \textlnot                 = {300,   },
⟨pmn|ugm⟩     \textlnot                 = {200,   },
⟨cmr⟩     \textregistered           = {100,   },
⟨pad⟩     \textregistered           = {200,100},
⟨pmn⟩     \textregistered           = { 50,150},
⟨ugm⟩     \textregistered           = {300,   },
⟨pmn⟩     \textasciimacron          = {150,200},
⟨cmr|pad⟩     \textdegree               = {500,100},
⟨pmn⟩     \textdegree               = {150,150},
⟨ugm⟩     \textdegree               = {300,200},
⟨cmr⟩     \textpm                   = {150,100},
⟨pad⟩     \textpm                   = {200,150},
⟨pmn|ugm⟩     \textpm                   = {150,200},
⟨cmr⟩     \textonesuperior          = {400,   },
⟨pad⟩     \textonesuperior          = {300,100},
⟨pmn⟩     \textonesuperior          = {200,100},
⟨ugm⟩     \textonesuperior          = {300,300},
⟨cmr⟩     \texttwosuperior          = {400,   },
⟨pad⟩     \texttwosuperior          = {300,   },
⟨pmn⟩     \texttwosuperior          = {200,100},
⟨ugm⟩     \texttwosuperior          = {300,200},
⟨cmr⟩     \textthreesuperior        = {400,   },
⟨pad⟩     \textthreesuperior        = {300,   },
⟨pmn⟩     \textthreesuperior        = {200,100},
⟨ugm⟩     \textthreesuperior        = {300,200},
⟨ugm⟩     \textmu                   = {   ,100},
⟨pmn⟩     \textasciiacute           = {300,200},
⟨cmr⟩     \textparagraph            = {200,   },
⟨pmn⟩     \textparagraph            = {   ,100},
⟨cmr⟩     \textperiodcentered       = {500,500},
⟨pad|pmn|ugm⟩     \textperiodcentered       = {300,400},
⟨cmr⟩     \textordmasculine         = {100,100},
⟨pmn⟩     \textordmasculine         = {200,200},
⟨ugm⟩     \textordmasculine         = {300,200},
⟨cmr⟩     \texteuro                 = {200,   },
⟨pad⟩     \texteuro                 = {100,   },
⟨pmn⟩     \texteuro                 = {100,-50},
⟨cmr⟩     \texttimes                = {200,200},
⟨pad⟩     \texttimes                = {200,100},
⟨pmn⟩     \texttimes                = { 70,100},
⟨ugm⟩     \texttimes                = {200,300},
⟨cmr|pad⟩     \textdiv                  = {200,200}
⟨pmn⟩     \textdiv                  = {150,200}
⟨ugm⟩     \textdiv                  = {200,300},
⟨ugm⟩     \textsection              = {   ,200},
⟨ugm⟩     \textonehalf              = { 50,100},
⟨ugm⟩     \textonequarter           = { 50,100},
⟨ugm⟩     \textthreequarters        = { 50,100},
⟨ugm⟩     \textsurd                 = {   ,100}
  }

⟨/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}
```

```
⟨*cmr⟩
\SetProtrusion
  [ name     = cmr-math-letters ]
  { encoding = OML,
    family   = cmm,
    series   = {m,b},
    shape    = it    }
  {
      A = {100, 50}, % \mathnormal
      B = { 50,   },
      C = { 50,   },
      D = { 50, 50},
      E = { 50,   },
      F = {100, 50},
      G = { 50, 50},
      H = { 50, 50},
      I = { 50, 50},
      J = {150, 50},
      K = { 50,100},
      L = { 50, 50},
      M = { 50,   },
      N = { 50,   },
      O = { 50,   },
      P = { 50,   },
      Q = { 50, 50},
      R = { 50,   },
      S = { 50,   },
      T = { 50,100},
      U = { 50, 50},
      V = {100,100},
      W = { 50,100},
      X = { 50,100},
      Y = {100,100},
      f = {100,100},
      h = {   ,100},
      i = {   , 50},
      j = {   , 50},
      k = {   , 50},
      r = {   , 50},
      v = {   , 50},
      w = {   , 50},
      x = {   , 50},
    "0B = { 50,100}, % \alpha
    "0C = { 50, 50}, % \beta
    "0D = {200,150}, % \gamma
    "0E = { 50, 50}, % \delta
    "0F = { 50, 50}, % \epsilon
    "10 = { 50,150}, % \zeta
    "12 = { 50,   }, % \theta
    "13 = {   ,100}, % \iota
    "14 = {   ,100}, % \kappa
    "15 = {100, 50}, % \lambda
    "16 = {   , 50}, % \mu
    "17 = {   , 50}, % \nu
    "18 = {   , 50}, % \xi
    "19 = { 50,100}, % \pi
    "1A = { 50, 50}, % \rho
    "1B = {   ,150}, % \sigma
    "1C = { 50,150}, % \tau
    "1D = { 50, 50}, % \upsilon
    "1F = { 50,100}, % \chi
    "20 = { 50, 50}, % \psi
    "21 = {   , 50}, % \omega
    "22 = {   , 50}, % \varepsilon
    "23 = {   , 50}, % \vartheta
    "24 = {   , 50}, % \varpi
    "25 = {100,   }, % \varrho
    "26 = {100,100}, % \varsigma
    "27 = { 50, 50}, % \varphi
    "28 = {100,100}, % \leftharpoonup
    "29 = {100,100}, % \leftharpoondown
    "2A = {100,100}, % \rightharpoonup
    "2B = {100,100}, % \rightharpoondown
    "2C = {300,200}, % \lhook
    "2D = {200,300}, % \rhook
    "2E = {   ,100}, % \triangleright
    "2F = {100,   }, % \triangleleft
    "3A = {   ,500}, % ., \ldotp
    "3B = {   ,500}, % ,
    "3C = {200,100}, % <
    "3D = {300,400}, % /
    "3E = {100,200}, % >
    "3F = {200,200}, % \star
    "5B = {   ,100}, % \flat
    "5E = {200,200}, % \smile
    "5F = {200,200}, % \frown
    "7C = {100,   }, % \jmath
    "7D = {   ,100}  % \wp
```

Remaining slots in the source file.

```
  }

```

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}
```

```
\SetProtrusion
   [ name     = cmr-math-symbols ]
   { encoding = OMS,
     family   = cmsy,
     series   = {m,b},
     shape    = n  }
   {
       A = {150, 50}, % \mathcal
       C = {   ,100},
       D = {   , 50},
       F = { 50,150},
       I = {   ,100},
       J = {100,150},
       K = {   ,100},
       L = {100,   },
       M = { 50, 50},
       N = { 50,100},
       P = {   , 50},
       Q = { 50,   },
       R = {   , 50},
       T = { 50,150},
       V = { 50, 50},
       W = {   , 50},
       X = {100,100},
       Y = {100,   },
       Z = {100,150},
     "00 = {300,300}, % -
     "01 = {   ,700}, % \cdot, \cdotp
     "02 = {150,250}, % \times
     "03 = {150,250}, % *, \ast
     "04 = {200,300}, % \div
     "05 = {150,250}, % \diamond
     "06 = {200,200}, % \pm
     "07 = {200,200}, % \mp
     "08 = {100,100}, % \oplus
     "09 = {100,100}, % \ominus
     "0A = {100,100}, % \otimes
     "0B = {100,100}, % \oslash
     "0C = {100,100}, % \odot
     "0D = {100,100}, % \bigcirc
     "0E = {100,100}, % \circ
     "0F = {100,100}, % \bullet
     "10 = {100,100}, % \asymp
     "11 = {100,100}, % \equiv
     "12 = {200,100}, % \subseteq
     "13 = {100,200}, % \supseteq
     "14 = {200,100}, % \leq
     "15 = {100,200}, % \geq
     "16 = {200,100}, % \preceq
     "17 = {100,200}, % \succeq
     "18 = {200,200}, % \sim
     "19 = {150,150}, % \approx
     "1A = {200,100}, % \subset
     "1B = {100,200}, % \supset
     "1C = {200,100}, % \ll
     "1D = {100,200}, % \gg
     "1E = {300,100}, % \prec
     "1F = {100,300}, % \succ
     "20 = {100,200}, % \leftarrow
     "21 = {200,100}, % \rightarrow
     "22 = {100,100}, % \uparrow
     "23 = {100,100}, % \downarrow
     "24 = {100,100}, % \leftrightarrow
     "25 = {100,100}, % \nearrow
     "26 = {100,100}, % \searrow
     "27 = {100,100}, % \simeq
     "28 = {100,100}, % \Leftarrow
     "29 = {100,100}, % \Rightarrow
     "2A = {100,100}, % \Uparrow
     "2B = {100,100}, % \Downarrow
     "2C = {100,100}, % \Leftrightarrow
     "2D = {100,100}, % \nwarrow
     "2E = {100,100}, % \swarrow
     "2F = {   ,100}, % \propto
     "30 = {   ,400}, % \prime
     "31 = {100,100}, % \infty
     "32 = {150,100}, % \in
     "33 = {100,150}, % \ni
     "34 = {100,100}, % \triangle, \bigtriangleup
     "35 = {100,100}, % \bigtriangledown
     "38 = {   ,100}, % \forall
     "39 = {100,   }, % \exists
     "3A = {200,   }, % \neg
     "3E = {200,200}, % \top
     "3F = {200,200}, % \bot, \perp
     "5E = {100,200}, % \wedge
     "5F = {100,200}, % \vee
     "60 = {   ,300}, % \vdash
     "61 = {300,   }, % \dashv
     "62 = {100,100}, % \lfloor
     "63 = {100,100}, % \rfloor
     "64 = {100,100}, % \lceil
     "65 = {100,100}, % \rceil
     "66 = {150,   }, % \lbrace
     "67 = {   ,150}, % \rbrace
     "68 = {400,   }, % \langle
     "69 = {   ,400}, % \rangle
     "6C = {100,100}, % \updownarrow
     "6D = {100,100}, % \Updownarrow
     "6E = {100,300}, % \, \backslash, \setminus
     "72 = {100,100}, % \nabla
     "79 = {200,200}, % \dagger
     "7A = {100,100}, % \ddagger
     "7B = {100,   }, % \mathparagraph
     "7C = {100,100}, % \clubsuit
     "7D = {100,100}, % \diamondsuit
     "7E = {100,100}, % \heartsuit
     "7F = {100,100}  % \spadesuit
```

Remaining slots in the source file.

```
   }

```

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}
```

```
⟨/cmr⟩
⟨/cfg-t⟩
```

15.8.7 AMS symbols

Settings for the AMS math fonts (`amssymb`).

```
⟨*cfg-u⟩
```

Symbol font ‘a’.

```
⟨*msa⟩
\SetProtrusion
   [ name     = AMS-a ]
   { encoding = U,
     family   = msa  }
   {
     "05  =  {150,250},  % \centerdot
     "06  =  {100,100},  % \lozenge
     "07  =  { 50, 50},  % \blacklozenge
     "08  =  { 50, 50},  % \circlearrowright
     "09  =  { 50, 50},  % \circlearrowleft
     "0A  =  {100,100},  % \rightleftharpoons
     "0B  =  {100,100},  % \leftrightharpoons
     "0D  =  {-50,200},  % \Vdash
     "0E  =  {-50,200},  % \Vvdash
     "0F  =  {-70,150},  % \vDash
     "10  =  {100,150},  % \twoheadrightarrow
     "11  =  {100,150},  % \twoheadleftarrow
     "12  =  { 50,100},  % \leftleftarrows
     "13  =  { 50, 80},  % \rightrightarrows
     "14  =  {120,120},  % \upuparrows
     "15  =  {120,120},  % \downdownarrows
     "16  =  {200,200},  % \upharpoonright
     "17  =  {200,200},  % \downharpoonright
     "18  =  {200,200},  % \upharpoonleft
     "19  =  {200,200},  % \downharpoonleft
     "1A  =  { 80,100},  % \rightarrowtail
     "1B  =  { 80,100},  % \leftarrowtail
     "1C  =  { 50, 50},  % \leftrightarrows
     "1D  =  { 50, 50},  % \rightleftarrows
     "1E  =  {250,   },  % \Lsh
     "1F  =  {   ,250},  % \Rsh
     "20  =  {100,100},  % \rightsquigarrow
     "21  =  {100,100},  % \leftrightsquigarrow
     "22  =  {100, 50},  % \looparrowleft
     "23  =  { 50,100},  % \looparrowright
     "24  =  { 50, 80},  % \circeq
     "25  =  {   ,100},  % \succsim
     "26  =  {   ,100},  % \gtrsim
     "27  =  {   ,100},  % \gtrapprox
     "28  =  {150, 50},  % \multimap
     "2B  =  {100,150},  % \doteqdot
     "2C  =  {100,150},  % \triangleq
     "2D  =  {100, 50},  % \precsim
     "2E  =  {100, 50},  % \lesssim
     "2F  =  { 50, 50},  % \lessapprox
     "30  =  {100, 50},  % \eqslantless
     "31  =  { 50, 50},  % \eqslantgtr
     "32  =  {100, 50},  % \curlyeqprec
     "33  =  { 50,100},  % \curlyeqsucc
     "34  =  {100, 50},  % \preccurlyeq
     "36  =  { 50,   },  % \leqslant
     "38  =  {   , 50},  % \backprime
     "39  =  {250,250},  % \dabar@ : the dash bar in \dash(left,right)arrow
     "3C  =  { 50,100},  % \succcurlyeq
     "3E  =  {   , 50},  % \geqslant
     "40  =  {   , 50},  % \sqsubset
     "41  =  { 50,   },  % \sqsupset
     "42  =  {   ,150},  % \vartriangleright, \rhd
     "43  =  {150,   },  % \vartriangleleft, \lhd
     "44  =  {   ,100},  % \trianglerighteq, \unrhd
     "45  =  {100,   },  % \trianglelefteq, \unlhd
     "46  =  {100,100},  % \bigstar
     "48  =  { 50, 50},  % \blacktriangledown
     "49  =  {   ,100},  % \blacktriangleright
     "4A  =  {100,   },  % \blacktriangleleft
     "4B  =  {   ,150},  % \dashrightarrow (the arrow)
     "4C  =  {150,   },  % \dashleftarrow
     "4D  =  { 50, 50},  % \vartriangle
     "4E  =  { 50, 50},  % \blacktriangle
     "4F  =  { 50, 50},  % \triangledown
     "50  =  { 50, 50},  % \eqcirc
     "56  =  {   ,150},  % \Rrightarrow
     "57  =  {150,   },  % \Lleftarrow
     "58  =  {100,300},  % \checkmark
     "5C  =  { 50, 50},  % \angle
     "5D  =  { 50, 50},  % \measuredangle
     "5E  =  { 50, 50},  % \sphericalangle
     "5F  =  {   , 50},  % \varpropto
     "60  =  {100,100},  % \smallsmile
     "61  =  {100,100},  % \smallfrown
     "62  =  { 50,   },  % \Subset
     "63  =  {   , 50},  % \Supset
     "66  =  {150,150},  % \curlywedge
     "67  =  {150,150},  % \curlyvee
     "68  =  { 50,150},  % \leftthreetimes
     "69  =  {100, 50},  % \rightthreetimes
     "6C  =  { 50, 50},  % \bumpeq
     "6D  =  { 50, 50},  % \Bumpeq
     "6E  =  {100,   },  % \lll
     "6F  =  {   ,100},  % \ggg
     "70  =  { 50,100},  % \ulcorner
     "71  =  {100, 50},  % \urcorner
     "75  =  {150,200},  % \dotplus
     "76  =  { 50,100},  % \backsim
     "78  =  { 50,100},  % \llcorner
     "79  =  {100, 50},  % \lrcorner
     "7C  =  {100,100},  % \intercal
     "7D  =  { 50, 50},  % \circledcirc
     "7E  =  { 50, 50},  % \circledast
     "7F  =  { 50, 50}   % \circleddash
```

Remaining slots in the source file.

```
   }

⟨/msa⟩
```

Symbol font ‘b’.

```
⟨*msb⟩
\SetProtrusion
   [ name     = AMS-b ]
   { encoding = U,
     family   = msb  }
   {
       A  =  { 50, 50},  % \mathbb
       C  =  { 50, 50},
       G  =  {   , 50},
       L  =  {   , 50},
       P  =  {   , 50},
       R  =  {   , 50},
       T  =  {   , 50},
       V  =  { 50, 50},
       X  =  { 50, 50},
       Y  =  { 50, 50},
     "00  =  { 50, 50},  % \lvertneqq
     "01  =  { 50, 50},  % \gvertneqq
     "02  =  { 50, 50},  % \nleq
     "03  =  { 50, 50},  % \ngeq
     "04  =  {100, 50},  % \nless
```

```
    "05  =  { 50,150},  % \ngtr
    "06  =  {100, 50},  % \nprec
    "07  =  { 50,150},  % \nsucc
    "08  =  { 50, 50},  % \lneqq
    "09  =  { 50, 50},  % \gneqq
    "0A  =  {100,100},  % \nleqslant
    "0B  =  {100,100},  % \ngeqslant
    "0C  =  {100, 50},  % \lneq
    "0D  =  { 50,100},  % \gneq
    "0E  =  {100, 50},  % \npreceq
    "0F  =  { 50,100},  % \nsucceq
    "10  =  { 50,   },  % \precnsim
    "11  =  { 50, 50},  % \succnsim
    "12  =  { 50, 50},  % \lnsim
    "13  =  { 50, 50},  % \gnsim
    "14  =  { 50, 50},  % \nleqq
    "15  =  { 50, 50},  % \ngeqq
    "16  =  { 50, 50},  % \precneqq
    "17  =  { 50, 50},  % \succneqq
    "18  =  { 50, 50},  % \precnapprox
    "19  =  { 50, 50},  % \succnapprox
    "1A  =  { 50, 50},  % \lnapprox
    "1B  =  { 50, 50},  % \gnapprox
    "1C  =  {150,200},  % \nsim
    "1D  =  { 50, 50},  % \ncong
    "1E  =  {100,150},  % \diagup
    "1F  =  {100,150},  % \diagdown
    "20  =  {100, 50},  % \varsubsetneq
    "21  =  { 50,100},  % \varsupsetneq
    "22  =  {100, 50},  % \nsubseteqq
    "23  =  { 50,100},  % \nsupseteqq
    "24  =  {100, 50},  % \subsetneqq
    "25  =  { 50,100},  % \supsetneqq
    "26  =  {100, 50},  % \varsubsetneqq
    "27  =  { 50,100},  % \varsupsetneqq
    "28  =  {100, 50},  % \subsetneq
    "29  =  { 50,100},  % \supsetneq
    "2A  =  {100, 50},  % \nsubseteq
    "2B  =  { 50,100},  % \nsupseteq
    "2C  =  { 50,100},  % \nparallel
    "2D  =  {100,150},  % \nmid
    "2E  =  {150,150},  % \nshortmid
    "2F  =  {100,100},  % \nshortparallel
    "30  =  {   ,150},  % \nvdash
    "31  =  {   ,150},  % \nVdash
    "32  =  {   ,100},  % \nvDash
    "33  =  {   ,100},  % \nVDash
    "34  =  {   ,100},  % \ntrianglerighteq
    "35  =  {100,   },  % \ntrianglelefteq
    "36  =  {100,   },  % \ntriangleleft
    "37  =  {   ,100},  % \ntriangleright
    "38  =  {100,200},  % \nleftarrow
    "39  =  {100,200},  % \nrightarrow
    "3A  =  {100,100},  % \nLeftarrow
    "3B  =  { 50,100},  % \nRightarrow
    "3C  =  {100,100},  % \nLeftrightarrow
    "3D  =  {100,200},  % \nleftrightarrow
    "3E  =  { 50, 50},  % \divideontimes
    "3F  =  { 50, 50},  % \varnothing
    "60  =  {200,   },  % \Finv
    "61  =  {   , 50},  % \Game
    "68  =  {100,100},  % \eqsim
    "69  =  { 50,   },  % \beth
    "6A  =  { 50,   },  % \gimel
    "6B  =  {150,   },  % \daleth
    "6C  =  {200,   },  % \lessdot
    "6D  =  {   ,200},  % \gtrdot
    "6E  =  {100,200},  % \ltimes
    "6F  =  {150,100},  % \rtimes
    "70  =  { 50,100},  % \shortmid
    "71  =  { 50, 50},  % \shortparallel
    "72  =  {200,300},  % \smallsetminus
    "73  =  {100,200},  % \thicksim
    "74  =  { 50,100},  % \thickapprox
    "75  =  { 50, 50},  % \approxeq
    "76  =  { 50,100},  % \succapprox
    "77  =  { 50, 50},  % \precapprox
    "78  =  {100,100},  % \curvearrowleft
    "79  =  { 50,150},  % \curvearrowright
    "7A  =  { 50,200},  % \digamma
    "7B  =  {100, 50},  % \varkappa
    "7F  =  {200,   }   % \backepsilon
```

Remaining slots in the source file.

```
  }

⟨/msb⟩
```

15.8.8 Euler

Euler Roman font (package `euler`).

```
⟨*eur⟩
\SetProtrusion
  [ name     = euler ]
  { encoding = U,
    family   = eur  }
  {
    "01  =  {100,100},
    "03  =  {100,150},
    "06  =  {   ,100},
    "07  =  {100,150},
    "08  =  {100,100},
    "0A  =  {100,100},
    "0B  =  {   , 50},
    "0C  =  {   ,100},
    "0D  =  {100,100},
    "0E  =  {   ,100},
    "0F  =  {100,100},
    "10  =  {100,100},
    "13  =  {   ,100},
    "14  =  {   ,100},
    "15  =  {   , 50},
    "16  =  {   , 50},
    "17  =  { 50,100},
    "18  =  { 50,100},
    "1A  =  {   , 50},
    "1B  =  {   , 50},
    "1C  =  { 50,100},
    "1D  =  { 50,100},
    "1E  =  { 50,100},
    "1F  =  { 50,100},
    "20  =  {   , 50},
    "21  =  {   , 50},
    "22  =  { 50,100},
    "24  =  {   , 50},
    "27  =  { 50,100},
      1  =  {100,100},
      7  =  { 50,100},
    "3A  =  {300,500},
    "3B  =  {200,400},
    "3C  =  {200,100},
    "3D  =  {200,200},
    "3E  =  {100,200},
      A  =  {   ,100},
      D  =  {   , 50},
      J  =  { 50,   },
      K  =  {   , 50},
      L  =  {   , 50},
      Q  =  {   , 50},
      T  =  { 50,   },
      X  =  { 50, 50},
      Y  =  { 50,   },
      h  =  {   , 50},
      k  =  {   , 50}
  }

```

Extended by the `eulervm` package.

```
\SetProtrusion
   [ name     = euler-vm,
     load     = euler ]
   { encoding = U,
     family   = zeur  }
   {
     "28  =  {100,200},
     "29  =  {100,200},
     "2A  =  {100,150},
     "2B  =  {100,150},
     "2C  =  {200,300},
     "2D  =  {200,300},
     "2E  =  {   ,100},
     "2F  =  {100,   },
     "3F  =  {150,150},
     "5B  =  {   ,100},
     "5E  =  {100,100},
     "5F  =  {100,100},
     "80  =  {   , 50},
     "81  =  {200,250},
     "82  =  {100,200}
   }

⟨/eur⟩
```

Euler Script font (`eucal`).

```
⟨*eus⟩
\SetProtrusion
   [ name     = euscript ]
   { encoding = U,
     family   = eus  }
   {
       A  =  {100,100},
       B  =  { 50,100},
       C  =  { 50, 50},
       D  =  { 50,100},
       E  =  { 50,100},
       F  =  { 50,   },
       G  =  { 50,   },
       H  =  {   ,100},
       K  =  {   , 50},
       L  =  {   ,150},
       M  =  {   , 50},
       N  =  {   , 50},
       O  =  { 50, 50},
       P  =  { 50, 50},
       T  =  {   ,100},
       U  =  {   , 50},
       V  =  { 50, 50},
       W  =  { 50, 50},
       X  =  { 50, 50},
       Y  =  { 50,   },
       Z  =  { 50,100},
     "00  =  {250,250},
     "18  =  {200,200},
     "3A  =  {200,150},
     "40  =  {   ,100},
     "5E  =  {100,100},
     "5F  =  {100,100},
     "66  =  { 50,   },
     "67  =  {   , 50},
     "6E  =  {200,200}
   }

\SetProtrusion
   [ name     = euscript-vm,
     load     = euscript ]
   { encoding = U,
     family   = zeus  }
   {
     "01  =  {600,600},
     "02  =  {200,200},
     "03  =  {200,200},
     "04  =  {200,200},
     "05  =  {150,150},
     "06  =  {200,200},
     "07  =  {200,200},
     "08  =  {100,100},
     "09  =  {100,100},
     "0A  =  {100,100},
     "0B  =  {100,100},
     "0C  =  {100,100},
     "0D  =  {100,100},
     "0E  =  {150,150},
     "0F  =  {100,100},
     "10  =  {150,150},
     "11  =  {100,100},
     "12  =  {150,100},
     "13  =  {100,150},
     "14  =  {150,100},
     "15  =  {100,150},
     "16  =  {200,100},
     "17  =  {100,200},
     "19  =  {150,150},
     "1A  =  {150,100},
     "1B  =  {100,150},
     "1C  =  {100,100},
     "1D  =  {100,100},
     "1E  =  {250,100},
     "1F  =  {100,250},
     "20  =  {150,200},
     "21  =  {150,200},
     "22  =  {150,150},
     "23  =  {150,150},
     "24  =  {100,200},
     "25  =  {150,150},
     "26  =  {150,150},
     "27  =  {100,100},
     "28  =  {100,100},
     "29  =  {100,150},
     "2A  =  {100,100},
     "2B  =  {100,100},
     "2C  =  {100,100},
     "2D  =  {150,150},
     "2E  =  {150,150},
     "2F  =  {100,100},
     "30  =  {100,100},
     "31  =  {100,100},
     "32  =  {100,100},
     "33  =  {100,100},
     "34  =  {100,100},
     "35  =  {100,100},
     "3E  =  {150,150},
     "3F  =  {150,150},
     "60  =  {   ,200},
     "61  =  {200,   },
     "62  =  {100,100},
     "63  =  {100,100},
     "64  =  {100,100},
     "65  =  {100,100},
     "68  =  {300,   },
     "69  =  {   ,300},
     "6C  =  {100,100},
     "6D  =  {100,100},
     "6F  =  {100,100},
     "72  =  {100,100},
     "73  =  {200,100},
     "76  =  {   ,100},
     "77  =  {100,   },
     "78  =  { 50, 50},
     "79  =  {100,100},
     "7A  =  {100,100},
     "7D  =  {150,150},
     "7E  =  {100,100},
     "A8  =  {100,100},
     "A9  =  {100,100},
     "AB  =  {200,200},
     "BA  =  {   ,200},
     "BB  =  {   ,200},
     "BD  =  {200,200},
     "DE  =  {200,200}
   }

⟨/eus⟩
```

Euler Fraktur font (`eufrak`).

```
⟨*euf⟩
\SetProtrusion
   [ name     = mathfrak ]
   { encoding = U,
     family   = euf  }
   {
       A  =  {   , 50},
       B  =  {   , 50},
       C  =  { 50, 50},
       D  =  {   , 80},
       E  =  { 50,   },
       G  =  {   , 50},
       L  =  {   , 80},
       O  =  {   , 50},
       T  =  {   , 80},
       X  =  { 80, 50},
       Z  =  { 80, 50},
       b  =  {   , 50},
       c  =  {   , 50},
       k  =  {   , 50},
       p  =  {   , 50},
       q  =  { 50,   },
       v  =  {   , 50},
       w  =  {   , 50},
       x  =  {   , 50},
       1  =  {100,100},
       2  =  { 80, 80},
       3  =  { 80, 50},
       4  =  { 80, 50},
       7  =  { 50, 50},
     "12  =  {500,500},
     "13  =  {500,500},
       !  =  {   ,200},
       '  =  {200,300},
       (  =  {200,   },
       )  =  {   ,200},
       *  =  {200,200},
       +  =  {200,250},
       -  =  {200,200},
      {,} =  {300,300},
       .  =  {400,400},
      {=} =  {200,200},
       :  =  {   ,200},
       ;  =  {   ,200},
       ]  =  {   ,200}
   }

⟨/euf⟩
⟨/cfg-u⟩
```

15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages `eurosans`, `europs`), ITC Euro fonts (package `euroitc`) and `marvosym`[24]).

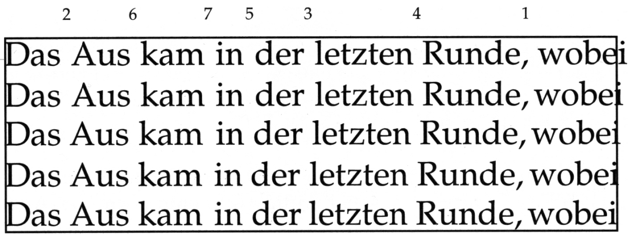
```
⟨*cfg-e⟩
\SetProtrusion
⟨zpeu|euroitc⟩   { encoding = U,
⟨mvs⟩   { encoding = {OT1,U},
⟨zpeu⟩      family   = zpeu }
⟨euroitc⟩      family   = {euroitc,euroitcs} }
⟨mvs⟩      family   = mvs }
   {
⟨zpeu⟩      E = {50, }
⟨euroitc⟩      E = {100,50}
⟨mvs⟩      164 = {50,50},   % \EUR
⟨mvs⟩      068 = {50,-100}  % \EURdig
   }

⟨*zpeu|euroitc⟩
\SetProtrusion
   { encoding = U,
⟨zpeu⟩      family   = zpeu,
⟨euroitc⟩      family   = {euroitc,euroitcs},
     shape    = it* }
   {
⟨zpeu⟩      E = {100,-50}
⟨euroitc⟩      E = {100,}
   }

⟨/zpeu|euroitc⟩
⟨*zpeu⟩
\SetProtrusion
```

---

[24] Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!



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

Figure 1:

Example of interword spacing (from: M. Siemoneit, *Typographisches Gestalten*, Frankfurt/M. 1989). The numbers indicate the preference for shrinking the interword space.

```
   { encoding = U,
     family   = {zpeus,eurosans} }
   {
     E = {100,50}
   }

\SetProtrusion
   { encoding = U,
     family   = {zpeus,eurosans},
     shape    = it* }
   {
     E = {200, }
   }

⟨/zpeu⟩
⟨/cfg-e⟩
```

15.9 Interword spacing

Default unit is space.

```
⟨*m-t|cmr⟩
%%% -----------------------------------------------------------------------
%%% INTERWORD SPACING

⟨/m-t|cmr⟩
⟨*m-t⟩
\SetExtraSpacing
   [ name = default ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
   {
```

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

```
      {,} = { ,-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)

```
          r  = { ,-300,300},
```

- [before or] after lowercase characters with ascenders

```
          b  = { ,-200,200},
          d  = { ,-200,200},
          f  = { ,-200,200},
          h  = { ,-200,200},
          k  = { ,-200,200},
          l  = { ,-200,200},
          t  = { ,-200,200},
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., ‘v’, or ‘w’

```
          c  = { ,-100,100},
          p  = { ,-100,100},
          v  = { ,-100,100},
          w  = { ,-100,100},
          z  = { ,-100,100},
          x  = { ,-100,100},
          y  = { ,-100,100},
```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```
          i  = { , 50, -50},
          m  = { , 50, -50},
          n  = { , 50, -50},
          u  = { , 50, -50},
```

- after colon and semicolon

```
          :  = { ,200,-200},
          ;  = { ,200,-200},
```

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

```
          .  = { ,250,-250},
          !  = { ,250,-250},
          ?  = { ,250,-250}
```

The order has to be reversed when enlarging is needed.’

```
   }

⟨/m-t⟩
```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a `factor` < 1000.)
- Should the first parameter also be used? (Probably.)
- What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTEX bug with spacing in combination with a non-zero `\spaceskip` (reported by *Axel Berger*):

```
\parfillskip0pt
\rightskip0pt plus 1em
\spaceskip\fontdimen2\font
 test test\par
\pdfadjustinterwordglue2
```

```
\stbscode\font`t=-50
 test test
\bye
```

Some more characters in T2A.[25]

```
⟨*cmr⟩
\SetExtraSpacing
   [ name     = T2A,
     load     = default ]
   { encoding = T2A,
     family   = cmr }
   {
      \cyrg  = { ,-300,300},
      \cyrb  = { ,-200,200},
      \cyrk  = { ,-200,200},
      \cyrs  = { ,-100,100},
      \cyrr  = { ,-100,100},
      \cyrh  = { ,-100,100},
      \cyru  = { ,-100,100},
      \cyrt  = { , 50, -50},
      \cyrp  = { , 50, -50},
      \cyri  = { , 50, -50},
      \cyrishrt  = { , 50, -50},
   }

```

15.9.1 Nonfrenchspacing

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the TEXbook:

> 'If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.'

The 'extra space' (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i.e., 333.

```
\SetExtraSpacing
   [ name     = nonfrench-cmr,
     load     = default,
     context  = nonfrench ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5},
     family   = cmr }
   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
  \sfcode`\. 3000
  \sfcode`\? 3000
  \sfcode`\! 3000
```

```
     . = {333,2000,-667},
     ? = {333,2000,-667},
     ! = {333,2000,-667},
```

```
  \sfcode`\: 2000
```

---

[25] Contributed by *Karl Karlsson*.

```
     : = {333,1000,-500},
```

```
\sfcode`\; 1500
```

```
     ; = {   , 500,-333},
```

```
\sfcode`\, 1250
```

```
    {,}= {   , 250,-200}
```

```
}
```

```
   }

⟨/cmr⟩
```

fontinst, however, which is also used to create the psnfss font metrics, sets `\fontdimen` 7 to 240 by default. Therefore, the fallback settings use this value for the first component.

```
⟨*m-t⟩
\SetExtraSpacing
   [ name     = nonfrench-default,
     load     = default,
     context  = nonfrench ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
   {
     . = {240,2000,-667},
     ? = {240,2000,-667},
     ! = {240,2000,-667},
     : = {240,1000,-500},
     ; = {   , 500,-333},
    {,}= {   , 250,-200}
   }

```

15.10 Additional kerning

Default unit is 1 em.

```
%%% -----------------------------------------------------------------------
%%% ADDITIONAL KERNING

```

A dummy list to be loaded when no context is active.

```
\SetExtraKerning
   [ name = empty ]
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
   { }

```

15.10.1 French

The ratio of `\fontdimen` 2 to `\fontdimen` 6 varies for different fonts, so that either the kerning of the colon (which should be a space, i.e., `\fontdimen` 2) or that of the other punctuation characters (TEX's `\thinspace`, i.e., one sixth of `\fontdimen` 6) may be inaccurate, depending on which `unit` we choose (`space` or `1em`). For Times, for example, a thin space would be 665. I don't know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a space. (Wikipedia[26] 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.)

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

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTEX.

```
\SetExtraKerning
   [ name     = french-guillemets,
     context  = french-guillemets,
     load     = french-default,
     unit     = space   ]
   { encoding = {T1,LY1} }
   {
    \guillemotleft  = { ,800}, % = 0.8\fontdimen2
    \guillemotright = {800, }
   }

\SetExtraKerning
   [ name     = french-guillemets-OT1,
     context  = french-guillemets,
     load     = french-default,
     unit     = space   ]
   { encoding = OT1     }
   { }

```

15.10.2 Turkish

```
\SetExtraKerning
   [ name     = turkish,
     context  = turkish ]
   { encoding = {OT1,T1,LY1} }
   {
     :  = {167, }, % = \thinspace
     !  = {167, },
    {=} = {167, }
   }

⟨/m-t⟩
⟨/config⟩
```

---

[26] http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

16 OpenType configuration files

These are the configuration files for the following OpenType fonts:[27]

- Latin Modern Roman
- Charis SIL[28]
- Palatino Linotype[29]

The settings are typeset in the respective font.

16.1 Character inheritance

OpenType fonts may differ considerably in how complete their arsenal of glyphs is. Therefore, each font family should have their own inheritance settings.

```

%%% ----------------------------------------------------------------------
%%% INHERITANCE

%% for xetex (EU1) and luatex (EU2), resp. both (TU)
⟨*LatinModernRoman⟩
\DeclareCharacterInheritance
   { encoding = {EU1,EU2,TU},
     family   = Latin Modern Roman }
  { A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǻ,Ȁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ặ,
       A},  % Greek
   Æ = {Ǽ},
   B = {Ƀ,
       B},  % Greek
   C = {Ç,Ć,Ĉ,Ċ,Č},
   D = {Đ,Ď,Ð,Ḍ,Ḏ},
   E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,
       E},  % Greek
   G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ},
   H = {Ĥ,Ħ,Ḥ,Ḧ,Ḫ,
       H},  % Greek
   I = {Ì,Í,Î,Ï,Ĩ,Ī,Ĭ,Į,İ,Ȉ,Ỉ,Ị,
       I},  % Greek
   J = {Ĵ},
   K = {Ķ,
       K},  % Greek
   L = {Ĺ,Ļ,Ł,Ḷ}, % Ľ,Ŀ,Ḹ
   M = {M},  % Greek
   N = {Ñ,Ń,Ņ,Ň,Ṅ,Ṇ,
       N},  % Greek
   O = {Ò,Ó,Ô,Õ,Ö,Ø,Ō,Ŏ,Ő,Ọ,Ơ,Ǫ,Ǿ,Ȍ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ỡ,Ở,Ợ,
       O},  % Greek
   P = {P},  % Greek
   R = {Ŕ,Ŗ,Ř,Ȑ,Ṛ,Ṙ,Ṝ},
   S = {Ś,Ŝ,Ş,Š,Ș,Ṣ},
   T = {Ţ,Ť,Ț,Ṭ,Ṯ,
       T},  % Greek
   U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ư,Ȕ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
   W = {Ŵ,Ẁ,Ẃ,Ẅ},
   X = {X},  % Greek
   Y = {Ý,Ŷ,Ÿ,Ỵ,Ỷ,Ỹ},
   Z = {Ź,Ż,Ž,Ẓ,
```

---

[27] This is file `microtype-utf.dtx`.

[28] Available at http://software.sil.org/charis.

[29] These settings have been contributed by *Loren B. Davis*.

```
          Z},  % Greek
    a = {à,á,â,ã,ä,å,ā,ă,ą,ǻ,ȁ,ạ,ả,ấ,ầ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ},
    æ = {ǽ},
    c = {ç,ć,ĉ,ċ,č},
    d = {đ,ḍ,ḏ},
    e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ẹ,ẻ,ẽ,ế,ề,ể,ễ,ệ},
    f = {/f_f},
    g = {ĝ,ğ,ġ,ģ,ǧ,ǵ},
    h = {ĥ,ħ,ḥ,ḧ,ḫ},
    i = {ì,í,î,ï,ĩ,ī,ĭ,į,ı,ȉ,ỉ,ị},
    j = {ĵ},
    k = {ķ},
    l = {ĺ,ļ,ł,ḷ,ḹ}, % ľ,l ·
    n = {ñ,ń,ņ,ň,ṅ,ṇ},
    o = {ò,ó,ô,õ,ö,ø,ō,ŏ,ő,ọ,ơ,ǫ,ǿ,ȍ,ỏ,ố,ồ,ổ,ỗ,ộ,ớ,ờ,ỡ,ở,ợ},
    r = {ŕ,ŗ,ř,ȑ,ṛ,ṙ,ṝ},
    s = {ś,ŝ,ş,š,ș,ṣ},
    t = {ţ,ț,ṭ,ṯ,ẗ}, % ť
    u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ư,ȕ,ụ,ủ,ứ,ừ,ử,ữ,ự},
    w = {ŵ,ẁ,ẃ,ẅ},
    y = {ý,ŷ,ÿ,ỳ,ỵ,ỷ,ỹ},
    z = {ź,ż,ž,ẓ},
  }
</LatinModernRoman>
<*CharisSIL>
\DeclareCharacterInheritance
   { encoding = {EU1,EU2,TU},
     family   = Charis SIL }
  { A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ȧ,Ǻ,Ȁ,Ȃ,Ḁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ꜳ,
         А,Ӑ,Ӓ}, % Cyrillic
    Æ = {Ǣ,
         Ǽ,Ӕ}, % Cyrillic
    B = {Ḃ,Ḅ,Ḇ,
         В}, % Cyr
    C = {Ç,Ć,Ĉ,Ḉ,Ċ,Č,
         С,Ҫ}, % Cyr
    D = {Ď,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ,Đ},
    E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ȩ,Ȇ,Ḕ,Ḗ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,Ể,Ễ,Ệ,
         Е,Ѐ,Ё,Ё̆}, % Cyr
    F = {Ḟ},
    G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ,Ḡ,G},
    H = {Ĥ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ,
         Н,Ң,Ҥ,Ӈ,Ӊ}, % Cyr
    I = {Ì,Í,Î,Ȉ,Ï,Į,Ĩ,Ị,Ī,İ,Ǐ,Ỉ,Ȋ,Ĭ,Ḭ,Ï,
         І,Ї,Ӏ,ӏ}, % Cyr
    J = {Ĵ,
         Ј}, % Cyr
    K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ,
         К,Ќ,Қ,Ҝ,Ҟ,Ҡ,Ӄ,Ԟ}, % Cyr
    L = {Ĺ,Ļ,Ľ,Ḷ,Ḹ,Ḻ,Ḽ}, % Ŀ
    M = {Ḿ,Ṁ,Ṃ,
         М,Ӎ}, % Cyr
    N = {Ñ,Ń,Ņ,Ň,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ,
         И,Й,Ѝ,Ҋ,Ӣ,Ӥ}, % Cyr
    O = {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Ǒ,Ǫ,Ǭ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ṍ,Ṏ,Ṑ,Ṓ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ơ,
         О,Ѳ,Ӧ,Ө,Ӫ, % Cyr
         Θ}, % Greek
    P = {Ṕ,Ṗ,
         Р,Ҏ},  % Cyr
    Q = {Ԛ}, % Cyr
    R = {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ},
    S = {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ,
         Ѕ}, % Cyr
```

```
 T = {Ţ,Ť,Ț,Ṫ,Ṭ,Ṯ,Ṱ,
     Т,Ҭ}, % Cyr
 U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
 V = {Ṽ,Ṿ},
 W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ,
     Ԝ}, % Cyr
 X = {Ẋ,Ẍ,
     Х,Ҳ,Ӽ,Ӿ},  % Cyr
 Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ,
     Ү,Ұ}, % Cyr
 Z = {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ},
 a = {à,á,â,ã,ä,å,ā,ă,ą,ǎ,ǟ,ǡ,ȧ,ǻ,ȁ,ȃ,ḁ,ạ,ả,ầ,ấ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ,ặ, % aʾ
     а,ӑ,ӓ}, % Cyr
 æ = {ǽ,
     ӕ}, % Cyr
 b = {ḃ,ḅ,ḇ},
 c = {ç,ć,ĉ,ċ,č,ḉ,
     с,ҫ}, % Cyr
 d = {ď,ḋ,ḍ,ḏ,ḑ,ḓ},
 e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ȩ,ȇ,ḕ,ḗ,ḙ,ḛ,ḝ,ẹ,ẻ,ẽ,ề,ế,ể,ễ,ệ,
     е,ѐ,ё,ӗ}, % Cyr
 f = {ḟ,ff}, % /f_f
 g = {ĝ,ğ,ġ,ģ,ǧ,ǵ,ḡ},
 h = {ĥ,ȟ,ḣ,ḥ,ḧ,ḩ,ḫ,ẖ,
     ћ,һ}, % Cyr
 i = {ı,ì,í,î,ȉ,ǐ,ï,į,ī,ị,ĩ,ḭ,ḯ,ỉ,ȋ,ĭ,
     і,ї}, % Cyr
 j = {ĵ,ǰ,
     ј}, % Cyr
 k = {ķ,ǩ,ḱ,ḳ,ḵ},
 l = {ĺ,ļ,ḻ,ḷ,ḹ,ḽ}, % ľ,ŀ
 m = {ḿ,ṁ,ṃ},
 n = {ñ,ń,ņ,ň,ǹ,ṅ,ṇ,ṉ,ṋ}, % ’n
 o = {ò,ó,ô,õ,ö,ō,ŏ,ő,ǒ,ǫ,ǭ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ṍ,ṏ,ṑ,ṓ,ọ,ỏ,ồ,ố,ổ,ỗ,ộ,ớ,ờ,ở,ỡ,ợ,
     о,ө,ӧ,ө,ӫ}, % Cyr
 p = {ṕ,ṗ,
     р,ҏ}, % Cyr
 q = {ԛ}, % Cyr
 r = {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
 s = {ś,ŝ,ş,š,ș,ṡ,ṣ,ṥ,ṧ,ṩ,
     ѕ}, % Cyr
 t = {ţ,ț,ṫ,ṭ,ṯ,ṱ,ẗ}, % ť
 u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự},
 v = {ṽ,ṿ},
 w = {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ,
     ԝ},  % Cyr
 x = {ẋ,ẍ,
     х,ҳ}, % Cyr
 y = {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,ỵ,ỷ,ỹ,
     у,ў,ӯ,ӱ,ӳ}, % Cyr
 z = {ź,ż,ž,ẑ,ẓ,ẕ},
% Cyrillic
 Г = {Ѓ,Ґ,Ғ,Ӷ,Ӻ},
 Ж = {Җ,Ӝ,Ӂ},
 З = {Ӟ,Ҙ},
 Л = {Ӆ},
 П = {Ԥ},
 У = {Ў,Ӯ,Ӱ,Ӳ},
 Ч = {Ҷ,Ҹ,Ӌ,Ӵ},
 Ы = {Ӹ},
 Ә = {Ӛ},
 Ҽ = {Ҿ},
 г = {ѓ,ґ,ғ,ӷ,ӻ},
 ж = {җ,ӂ,ӝ},
    з = {ҙ,ӟ},
    и = {й,ѝ,ҋ,ӣ,ӥ},
    к = {ќ,қ,ҝ,ҟ,ҡ,ӄ,ԟ},
    л = {ӆ},
    м = {ӎ},
    н = {ң,ҥ,ӈ,ӊ},
    п = {ԥ},
    т = {ҭ},
    х = {ҳ,ӿ},
    ч = {ҷ,ҹ,ӌ,ӵ},
    ш = {щ},
    ы = {ӹ},
    э = {ӭ},
    ҽ = {ҿ},
    ә = {ӛ},
    ү = {ұ},
    Г = {Γ}, % Greek
    П = {Π}, % Greek
  }

  % missing: tipa, math, symbols, ...
⟨/CharisSIL⟩
⟨*PalatinoLinotype⟩
\DeclareCharacterInheritance
  { encoding = {EU1,EU2,TU},
    family = {PalatinoLinotype} }
```

Unfortunately, I don’t have a Palatino variant containing all of the following glyphs. The settings are typeset in TEX Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as ‘�’. To see the real settings, consult `mt-PalatinoLinotype.cfg`.

```
  { A = {À,Á,Â,Ã,Ä,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ȧ,Ǻ,Ȁ,Ȃ,Ḁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ꜳ},
    B = {Ḃ,Ḅ,Ḇ},
    C = {Ç,Ć,Ĉ,Ḉ,Ċ,Č},
    D = {Ď,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ},
    E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ȩ,Ȇ,Ḕ,Ḗ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,Ể,Ễ,Ệ},
    F = {Ḟ},
    G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ,Ḡ},
    H = {Ĥ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ},
    I = {Ì,Í,Î,Ȉ,Ï,Į,Ĩ,Ị,Ī,İ,Ǐ,Ỉ,Ȋ,Ĭ,Ḭ,Ï},
    J = {Ĵ},
    K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ},
    L = {Ĺ,Ļ,Ľ,Ḷ,Ḹ,Ḻ,Ḽ,Ỻ,Ŀ,Ł}, % L·
    M = {Ḿ,Ṁ,Ṃ},
    N = {Ñ,Ń,Ņ,Ň,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ},
    O = {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Ǒ,Ǫ,Ǭ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ṍ,Ṏ,Ṑ,Ṓ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ},
    P = {Ṕ,Ṗ},
    R = {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ},
    S = {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ},
    T = {Ţ,Ť,Ț,Ṫ,Ṭ,Ṯ,Ṱ},
    U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
    V = {Ṽ,Ṿ},
    W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ},
    X = {Ẋ,Ẍ},
    Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ},
    Z = {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ},
    a = {à,á,â,ã,ä,å,ā,ă,ą,ǎ,ǟ,ǡ,ȧ,ǻ,ȁ,ȃ,ḁ,ạ,ả,ầ,ấ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ,ặ}, % aᵓ
    b = {ḃ,ḅ,ḇ},
    c = {ç,ć,ĉ,ċ,č,ḉ},
    d = {ď,ḋ,ḍ,ḏ,ḑ,ḓ},
    e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ȩ,ȇ,ḕ,ḗ,ḙ,ḛ,ḝ,ẹ,ẻ,ẽ,ề,ế,ể,ễ,ệ},
    f = {ḟ,ﬀ},
    g = {ĝ,ğ,ġ,ģ,ǧ,ǵ,ḡ},
    h = {ĥ,ȟ,ḣ,ḥ,ḧ,ḩ,ḫ,ẖ},
    i = {ı,ì,í,î,ȉ,ǐ,ï,į,ī,ị,ĩ,ḭ,ḯ,ỉ,ȋ,ĭ},
    j = {ĵ,ǰ},
    k = {ķ,ǩ,ḱ,ḳ,ḵ},
    l = {ĺ,ļ,ḻ,ḷ,ḹ,ḽ}, % ľ,ŀ
    m = {ḿ,ṁ,ṃ},
    n = {ñ,ń,ņ,ň,ǹ,ṅ,ṇ,ṉ,ṋ}, % ŉ
    o = {ò,ó,ô,õ,ö,ō,ŏ,ő,ǒ,ǫ,ǭ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ṍ,ṏ,ṑ,ṓ,ọ,ỏ,ồ,ố,ổ,ỗ,ộ,ớ,ờ,ở,ỡ,ợ},
    p = {ṕ,ṗ},
    r = {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
    s = {ś,ŝ,ş,š,ș,ṡ,ṣ,ṥ,ṧ,ṩ},
    t = {ţ,ț,ṫ,ṭ,ṯ,ṱ,ẗ}, % ť
    u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự},
    v = {ṽ,ṿ},
    w = {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ},
    x = {ẋ,ẍ},
    y = {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,ỵ,ỷ,ỹ},
    z = {ź,ż,ž,ẑ,ẓ,ẕ},
  }
⟨/PalatinoLinotype⟩
```

16.2 Character protrusion

```

%%% ------------------------------------------------------------------------
%%% PROTRUSION

⟨*LatinModernRoman⟩
\SetProtrusion
   [ name     = LMR-default ]
   { encoding = {EU1,EU2,TU},
     family   = Latin Modern Roman }
   {
     A = {50,50},
     Æ = {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 = { ,70},
     v = {50,50},
     w = {50,50},
     x = {50,50},
     y = {50,70},
     0 = { ,50},
     1 = {100,200},
     2 = {50,50},
     3 = {50,50},
     4 = {70,70},
     5 = { ,50},
     6 = { ,50},
     7 = {50,100},
     8 = { ,50},
     9 = { ,50},
     . = { ,700},
    {,}= { ,500},
     : = { ,500},
     ; = { ,500},
     ! = { ,100},
     ? = { ,200},
     @ = {50,50},
     ~ = {200,250},
    \% = {50,50},
     * = {300,300},
     + = {250,250},
     - = {400,500},  % /hyphen
     – = {400,300},  % /endash
     — = {300,200},  % /emdash
     _ = {200,200},  % /underscore
     / = {200,300},
    /backslash    = {200,300},
     ' = {300,400}, % /quotesingle
     ‘ = {500,700},  ’ = {500,600},
     “ = {500,300},  ” = {200,600},
     , = {400,400},  „ = {400,400},
     ‹ = {400,400},  › = {300,500},
     « = {300,200},  » = {100,400},
     ¡ = {100,   },  ¿ = {100,   },
     ( = {300,   },  ) = {   ,300},
     <  = {200,100}, > = {100,200},
    /braceleft = {400,200}, /braceright = {200,400},
    /angleleft = {400,   }, /angleright = {   ,400},
     † = {100,100},
     ‡ = { 80, 80},
     • = {200,200},
      · = {400,450}, % / periodcentered
     ℃ = { 80, 50},
     ₵ = {   , 50},
     ° = {400,400},
     ™ = {100,200},
     © = {100,100},
     ® = {100,100},
     ª = {100,200},
     º = {100,200},
     ¹ = {200,250},
     ² = { 50,100},
     ³ = { 50,100},
     ¬ = {200,   },
     − = {300,300},
     ± = {150,200},
     × = {150,250},
     ÷ = {150,250},
     € = {100,   },
    /one.oldstyle   = {100,100},
    /two.oldstyle   = { 50, 50},
    /three.oldstyle = { 30, 80},
    /four.oldstyle  = { 50, 50},
    /seven.oldstyle = { 50, 80},
     Γ = {   ,180}, % /Gamma
     Δ = {100,100}, % /Delta
     Θ = { 50, 50}, % /Theta
     Λ = {100,100}, % /Lambda
%      Ξ = {,},        % /Xi
%      Π = {,},        % /Pi
     Σ = { 50, 50}, % /Sigma
     ϒ = {100,100}, % /Upsilon
     Φ = { 50, 50}, % /Phi
     Ψ = { 50, 50}, % /Psi
%      Ω = {,},        % /Omega

    }
```

```

\SetProtrusion
   [ name     = LMR-it ]
   { encoding = {EU1,EU2,TU},
     family   = Latin Modern Roman,
     shape    = {it,sl}       }
   {
    A = {125,100},
    Æ = {125,-55},
    B = {90,-40},
    C = {145,-75},
    D = {75, -28},
    E = {80,-55},
    F = {85,-80},
    G = {153,-15},
    H = {73,-60},
    I = {140,-120},
    IJ = {140,-80},
    J = {135,-80},
    K = {70,-30},
    L = {87, 40},
    M = {67,-45},
    N = {75,-55},
    O = {150,-30},
    Œ = {150,-55},
    P = {82,-50},
    Q = {150,-30},
    R = {75, 15},
    S = {90,-65},
    $ = {100,-20},
    T = {220,-85},
    U = {230,-55},
    V = {260,-60},
    W = {185,-55},
    X = {70,-30},
    Y = {250,-60},
    Z = {90,-60},
    a = {150,-10},
    b = {170,   },
    c = {173,-10},
    d = {150,-55},
    e = {180, },
    f = {   ,-250},
    g = {150,-10},
    h = {100, },
    i = {210, },
    ij = {210,-40},
    j = {   ,-40},
    k = {110,-50},
    l = {240,-110},
    m = {80, },
    n = {115, },
    o = {155, },
    q = {170,-40},
    r = {155,-40},
    s = {130, },
    t = {230,-10},
    u = {120, },
    v = {140,-25},
    w = {98,-20},
    x = {65,-40},
    y = {130,-20},
    z = {110,-80},
    0 = {170,-85},
    1 = {230,110},
 2 = {130,-70},
 3 = {140,-70},
 4 = {130,80},
 5 = {160, },
 6 = {175,-30},
 7 = {250,-150},
 8 = {130,-40},
 9 = {155,-80},
 . = { ,500},
{,}= { ,450},
 : = { ,300},
 ; = { ,300},
 & = {130,30},
\% = {180,50},
 * = {380,20},
 + = {180,200},
 @ = {180,10},
 ~ = {200,150},
 ( = {300, },    ) = {  ,70},
 / = {100,100},
 - = {500,300}, % /hyphen
 – = {500,300}, % /endash
 — = {400,170}, % /emdash
 _ = {100,200}, % /underscore
 ' = {300,400}, % /quotesingle
 " = {500,300},
 ‘ = {800,200},  ’ = {800,-20},
 “ = {540,100},  ” = {500,100},
 , = {300,700},  „ = {200,600},
 ‹ = {500,300},  › = {400,400},
 « = {400,100},  » = {200,300},
 ¡ = {200,   },  ¿ = {200,   },
 < = {300,100},  > = {200,100},
/backslash   = {300,300},
/braceleft   = {400,100},   /braceright    = {200,200},
 † = {200, 80},
 ‡ = {120, 80},
 • = {220,100},
 · = {550,300}, % / periodcentered
 ℃ = {170,   },
 ₵ = {100, 50},
 ¶ = {200,   },
 ° = {500,300},
 ™ = {200, 70},
 © = { 50, 70},
 ® = { 50, 70},
 ª = {140,100},
 º = {140,100},
 ¹ = {400,150},
 ² = {250, 80},
 ³ = {250, 80},
 ¬ = {250, 80},
 − = {300,200},
 ± = {150,170},
 × = {200,200},
 ÷ = {200,200},
 € = {150,   },
/one.oldstyle   = {100,100},
/two.oldstyle   = {100, 80},
/three.oldstyle = { 80, 50},
/four.oldstyle  = { 80, 80},
/five.oldstyle  = { 50,   },
/six.oldstyle   = { 50,   },
/seven.oldstyle = { 80, 80},
/eight.oldstyle = { 50,   },
    Γ = {100,120}, % /Gamma
    Δ = {120,100}, % /Delta
    Θ = {120, 50}, % /Theta
    Λ = {130,100}, % /Lambda
    Ξ = {100,},     % /Xi
    Π = {100,},     % /Pi
    Σ = {100, 50}, % /Sigma
    Υ = {180,100}, % /Upsilon
    Φ = {130, 70}, % /Phi
    Ψ = {130, 50}, % /Psi
    Ω = { 50,},     % /Omega
   }
⟨/LatinModernRoman⟩
⟨*CharisSIL⟩
\SetProtrusion
   [ name     = Charis-default ]
   { encoding = {EU1,EU2,TU},
     family   = Charis SIL }
   {
   A = {50,50},
   Æ = {50,50},
   C = {50, },
   D = { ,50},
   F = { ,50},
   G = {50, },
   J = {100, },
   K = { ,50},
   L = { ,50},
   L· = { ,100},
   O = {50,50},
   Œ = {50, },
   P = { ,50},
   Q = {50,70},
   R = { ,50},
   ẞ = { ,40}, % capital sharp s
   T = {50,50},
   V = {50,50},
   W = {50,50},
   X = {50,50},
   Y = {50,50},
   k = { ,50},
   l· = { ,150},
   r = { ,50},
   t = { ,50},
   v = {50,50},
   w = {50,50},
   x = {50,50},
   y = { ,50},
   1 = {150,150},
   2 = {50,50},
   3 = {50, },
   4 = {100,50},
   6 = {50, },
   7 = {50,80},
   9 = {50,50},
   . = { ,600},
  {,}= { ,500},
   : = { ,400},
   ; = { ,300},
   ! = { ,100},
   ? = { ,200},
   @ = {50,50},
   ~ = {200,250},
  \% = { ,50},
   * = {300,300},
   + = {200,250},
   / = { ,200},
  /backslash = {150,200},
   | = {200,200},
   - = {400,500}, % hyphen
   – = {200,300}, % endash
   — = {150,250}, % emdash
   ― = {200,200}, % Horizontal Bar = \texttwelveudash
   ‒ = {150,150}, % Figure Dash = \textthreequartersemdash
   _ = {100,100},
  {=} = {100,100},
   ‘ = {300,400}, ’ = {300,400},
   “ = {300,300}, ” = {300,300},
   , = {400,400}, „ = {300,300},
   ‹ = {400,300}, › = {300,400},
   « = {200,200}, » = {150,300},
   ¡ = {100,   }, ¿ = {100,   },
   ( = {200,   }, ) = {   ,200},
   < = {200,150}, > = {100,200},
   [ = {100,   }, ] = {   ,100},
  /braceleft = {200,   }, /braceright = {   ,300},
   † = { 80, 80},
   ‡ = {100,100},
   • = {200,200},
   ° = {150,200},
   ™ = {150,150},
   ¢ = { 50,   },
   £ = { 50,   },
   ¦ = {200,200},
   © = {100,100},
   ® = {100,100},
   ª = {100,200},
   º = {200,200},
   ¬ = {200, 50},
   µ = {   ,100},
   ¶ = {   ,100},
   · = {300,400},
   ¹ = {200,300},
   ² = {100,200},
   ³ = {100,200},
   € = {100,   },
   ± = {150,200},
   × = {200,200},
   ÷ = {250,250},
  /minus = {200,200},
   − = {200,200},
  % Cyrillic
   Б = {  ,50},
   Г = {  ,130},
   Ж = {50,50},
   З = {30,50},
   Л = {50,  },
   У = {50,50},
   Ф = {50,50},
   Ч = {100, },
   Ъ = {  ,50},
   Ь = {  ,50},
   Э = {50,50},
   Ю = {  ,40},
   Я = {50,  },
   Ѵ = {50,50},
   Ҽ = {50,  },
   Ѣ = {50,100},
   Є = {50,  },
   Љ = {50,50},
    Њ = { ,50},
    Ћ = {50,50},
    Ԅ = {100,100},
    Ԇ = {50,50},
    Ѣ = { ,50},
    Ҍ = { ,50},
    Ԉ = {50,80},
    Ԋ = { ,80},
    Ԏ = {50,50},
    Ԓ = {50, },
    Ԕ = {50,40},
    Ԗ = { ,50},
    Ӕ = {50, },
    Ԡ = { ,50},
    Ԣ = { ,50},
    Ԃ = { ,100},
    б = {50,50},
    г = { ,70},
    к = { ,50},
    л = {50, },
    т = {50,50},
    ф = {50,50},
    ч = {50, },
    ъ = { ,50},
    ь = { ,50},
    э = { ,50},
    я = {50, },
    љ = {50, },
    њ = { ,50},
    ѣ = { ,50},
    ѵ = {50,50},
    ҽ = {50, },
    ҍ = { ,50},
    ү = {50,50},
    ҕ = { ,50},
    ҧ = { ,50},
    ԃ = { ,100},
    ԅ = {100,100},
    ԇ = {50,50},
    ԉ = {50,70},
    ԋ = { ,70},
    ӕ = {50,30},
    ԡ = { ,50},
    ԣ = { ,50},
    %    Д П Ц Ш Щ Ы Ҕ Ҧ Ꙍ Ә Ҵ Џ Ӭ Ӡ Ԑ Ԁ
    %    в д ж з и м н п ц ш ы ю ђ є ћ џ ә ԑ ꙍ ҵ ӡ ԁ ԍ ԏ ԓ ԕ ԗ
   % Greek
    Δ = {50,50},
    Ψ = {50,50},
    γ = {70,70},
    λ = {40,70},
    π = {40,50},
    ρ = { ,50},
    σ = { ,50},
    χ = {50,50},
}

\SetProtrusion
   [ name     = Charis-it   ]
   { encoding = {EU1,EU2,TU},
     family   = Charis SIL,
     shape    = {it,sl}  }
   {
    C = {50, },
    G = {50, },
    J = {50,  },
    L = {50,50},
    O = {50,  },
    Œ = {50,  },
    Q = {50,  },
    S = {50,  },
    $ = {50,  },
    T = {70,  },
    o = {50,50},
    p = {  ,50},
    q = {50,  },
    t = {  ,50},
    w = {  ,50},
    y = {  ,50},
    1 = {150,100},
    3 = {50,  },
    4 = {100, },
    6 = {50,  },
    7 = {100, },
    . = { ,700},
   {,}= { ,600},
    : = { ,400},
    ; = { ,400},
    ? = { ,150},
    & = {  ,80},
   \% = {50,50},
    * = {300,200},
   + = {250,250},
    @ = {80,50},
    ~ = {150,150},
    / = {  ,150},
   /backslash = {150,150},
    - = {300,400}, % hyphen
    – = {200,300}, % endash
    — = {150,200}, % emdash
    _ = {  ,100},
   {=} = {200,200},
   ± = {150,200},
   × = {250,250},
   ÷ = {250,250},
    ° = {150,200},
    · = {300,400},
    ‘ = {400,200},  ’ = {400,200},
    “ = {300,200},  ” = {400,200},
    , = {200,500},  „ = {150,500},
    ‹ = {300,400},  › = {200,500},
    « = {200,300},  » = {150,400},
    ( = {200,   },  ) = {   ,200},
   < = {200,200},  > = {200,200},
   /braceleft = {300,   }, /braceright = {   ,200},
  % Cyrillic
    Ж = {50,30},
    Л = {50,  },
    У = {50,30},
    Ф = {50,  },
    Ч = {100, },
    Ъ = {  ,50},
    Ь = {  ,50},
    Э = {50,50},
    Я = {50,  },
    Ѵ = {50,50},
    Љ = {50,50},
    Ԅ = {140,100},
    Ԇ = {70,50},
    Ԉ = {50,80},
    Ԋ = { ,80},
    Ԏ = {50,50},
    г = {50,50},
    д = {50,30},
    м = {50, },
    ф = {50, },
    ч = {50, },
    ъ = { ,50},
    ь = { ,50},
    э = { ,50},
    я = {50, },
    љ = {50,50},
    њ = { ,50},
    ѵ = {50,50},
    ҍ = { ,50},
    ԅ = {140,100},
    ԇ = {70,50},
    ԉ = {50,70},
    ԋ = { ,70},
   % Greek
    Γ = { ,130},
    Δ = {50,50},
    Ψ = {50,50},
    γ = {70,70},
    λ = {40,70},
    π = {40,50},
    ρ = { ,50},
    σ = { ,50},
    χ = {50,50},
    }
```

The small caps glyph names in Charis SIL have changed with version 5.0 of the font. We try to get the names right both with LuaTEX (where we can simply query the font version) and with XƎTEX (where we check for glyph name).

```

 % quick and dirty -- maybe we'll promote this to a
 % regular key some time
\define@key{MT@pr@c}{command}{\csname #1\endcsname}

 % glyph names have changed with version 5.0 of Charis SIL:
 % before: /a.SC, /b.SC, ...
 % after:  /a.sc, /b.sc, ...
\ifx\MT@lua\@undefined
  \gdef\MT@get@CHARIS@SC{
    % test whether glyph "a.sc" exists
    \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
      \gdef\MT@CHARIS@SC{sc}%
    \else
      \gdef\MT@CHARIS@SC{SC}%
    \fi
  }
\else
  \gdef\MT@get@CHARIS@SC{
    \gdef\MT@CHARIS@SC{\MT@lua{
      % check font version
% --  why doesn't this work?:
%     f = font.getfont(font.current());
%     i = fontloader.info(f.filename);
%     if (tonumber(i.version) < 5) then;
      if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
        tex.print("SC");
      else;
        tex.print("sc");
      end
    }}
  }
\fi

\SetProtrusion
   [ name     = Charis-sc,
     load     = Charis-default,
     command  = {MT@get@CHARIS@SC} ]
   { encoding = {EU1,EU2,TU},
     family   = Charis SIL,
     shape    = {sc}  }
   {
  % ᴀ = {100,100}, % etc., doesn't work with \textsc
   /a.\MT@CHARIS@SC = {100,100},
   /c.\MT@CHARIS@SC = {50,  },
   /d.\MT@CHARIS@SC = {  ,50},
   /f.\MT@CHARIS@SC = {  ,50},
   /g.\MT@CHARIS@SC = {50,  },
   /j.\MT@CHARIS@SC = {100, },
   /k.\MT@CHARIS@SC = {  ,50},
   /l.\MT@CHARIS@SC = {  ,50},
 /f_l.\MT@CHARIS@SC = {  ,50},
   /o.\MT@CHARIS@SC = {50,50},
  /oe.\MT@CHARIS@SC = {50,  },
   /q.\MT@CHARIS@SC = {50,70},
   /r.\MT@CHARIS@SC = {  ,50},
   /t.\MT@CHARIS@SC = {50,100},
   /v.\MT@CHARIS@SC = {50,50},
   /w.\MT@CHARIS@SC = {50,50},
   /x.\MT@CHARIS@SC = {50,50},
   /y.\MT@CHARIS@SC = {50,50}
   }
⟨/CharisSIL⟩
⟨*PalatinoLinotype⟩
\SetProtrusion
   [ name     = palatino-default ]
   { encoding = {EU1,EU2,TU},
     family   = {PalatinoLinotype} }
   {
   A = {50,50},
   D = {  ,50},
   J = {50,  },
   K = {  ,50},
   L = {  ,50},
   O = {25,  },
   T = {50,50},
   V = {50,50},
   W = {50,50},
   X = {50,50},
   Y = {50,50},
   b = {  ,25},
   d = {25,30},
   f = {  ,50},
   g = {  ,100},
   k = {  ,50},
   p = {  ,50},
   q = {50,  },
   r = {  ,50},
   t = {  ,50}, � = { ,50}, � = {  ,50},
   v = {75,50},
   w = {50,50},
   x = {50,50},
   y = {50,70},
   1 = {100,50},
  2 = {25,50},
  4 = {50,  },
  6 = {50,  },
  9 = {25,  },
  Æ = {100, },
  Œ = {25,  },
  . = { ,700},     ‥ = { ,350},    … = {,150},
 {,}= { ,500},
  : = { ,500},
  ; = { ,500},
  ! = { ,100},     ‼ = { ,100},
  ? = { ,200},     ‽ = { ,200},
  @ = {50,50},
  ~ = {200,250},
  & = {50,100},
 \% = {100,100},
  * = {200,200},
  + = {250,250},
  ( = {100,   },    ) = {   ,300},
  / = {200,300},
  - = {400,500},
  \textendash       = {300,300},  \textemdash        = {200,200},
  \textquoteleft    = {500,700},  \textquoteright    = {500,700},
  \textquotedblleft = {300,400},  \textquotedblright = {300,400},
  \textbackslash    = {200,300},
  \quotesinglbase   = {400,400},  \quotedblbase      = {400,400},
  \guilsinglleft    = {400,400},  \guilsinglright    = {300,500},
  \guillemotleft    = {300,300},  \guillemotright    = {200,400},
  \textexclamdown   = {100,   },  \textquestiondown  = {100,   },
  \textbraceleft    = {400,200},  \textbraceright    = {200,400},
  \textless         = {200,100},  \textgreater       = {100,200},
  ≤                 = {200,100},  ≥                  = {100,200},
  \textminus               = {300,300},
  \texttrademark           = {200,200},
  \textcopyright           = {200,200},
  \textregistered          = {200,200},
  \textdegree              = {300,300},
  ¦                 = {450,500},  ¬                  = {250,150},
  ⌐                 = {150,250},
  ·                        = {850, 700},
  ¶                        = {100,0},
  ×                        = {150, 300},
  ª                 = {300,300},  º                  = {300,300},
  ⁰ = {200,400},
  ¹ = {400,350},        ² = {200,300},          ³ = {250,400},
  ⁴ = {250,350},        ⁵ = {200,300},          ⁶ = {250,400},
  ⁷ = {200,450},        ⁸ = {250,400},          ⁹ = {200,350},
  ₀ = {200,400},
  ₁ = {400,250},        ₂ = {200,300},          ₃ = {250,400},
  ₄ = {250,350},        ₅ = {200,300},          ₆ = {250,400},
  ₇ = {200,450},        ₈ = {250,400},          ₉ = {200,350},
  ± = {150,100},                                ÷ = {300,300},
  þ = {  ,25},
  ₊ = {300,450},        ₋ = {300,450},
  ⁺ = {300,450},        ⁻ = {300,450},
  †                 = {200,250},  ‡                  = {200,250},
  π = {50,  },
  ſ = {  ,50},
  № = {100,150},
  \textservicemark         = {100,200},
  - = {400,500},        - = {400,500},          − = {200,300},
  – = {205,305},        — = {200,300},          ― = {50,150},
  • = {125,200},
%   /a.sc = {50,50},

   }
```

```

\SetProtrusion
   [ name     = palatino-it   ]
   { encoding = {EU1,EU2,TU},
     family   = {PalatinoLinotype},
     shape    = {it,sl}  }
   {
   A = {50,50},
   Æ = {50, },
   B = {50, },
   C = {50, },
   D = {50,50},
   E = {50, },
   F = {50, },
   G = {50, },
   H = {50, },
   K = {50, },
   L = {50, },
   O = {50, },
   Œ = {50, },
   P = {50, },
   Q = {50, },
   R = {50, },
   S = {50, },
   $ = {50, },
   T = {100, },
   U = {50, },
   V = {100,50},
   W = {50, },
   X = {50, },
   Y = {100,50},
   b = { ,50},
   c = {25, },
   g = {75, },
   i = {25, },
   m = { ,50},
   n = { ,50},
   p = { ,25},
   q = {25, },
   x = { ,50},
   1 = {100, },
   2 = {50, },
   4 = {50, },
   7 = {50, },
   . = { ,500},     ‥ = { ,350},   … = { ,200},
  {,}= { ,500},
   : = { ,300},
   ; = { ,300},
   ? = { ,300},     ‽ = { ,300},
   & = {50,50},
  \% = {100,100},
   * = {200,200},
   + = {150,200},
   @ = {50,50},
   ~ = {200,150},
   ( = {200, },    ) = { ,200},
   / = {100,200},
   - = {300,500},
   \textendash      = {300,300},  \textemdash       = {200,200},
   \textquoteleft   = {700,400},  \textquoteright   = {700,400},
   \textquotedblleft = {500,300},  \textquotedblright = {500,300},
   _ = {100,100},
   \textbackslash   = {100,200},
   \quotesinglbase  = {500,500},  \quotedblbase     = {400,400},
   \guilsinglleft   = {400,400},  \guilsinglright   = {300,500},
   \guillemotleft   = {300,300},  \guillemotright   = {300,300},
   \textexclamdown  = {100,  },  \textquestiondown = {200,  },
   \textbraceleft   = {200,100},  \textbraceright   = {200,200},
   \textless        = {300,100},  \textgreater      = {200,100},
   ≤                = {200,100},  ≥                 = {100,200},
   ¦                = {450,500},  ¬                 = {250,150},
   ·                   = {850, 700},
   ¶                   = {100,0},
   ×                   = {150, 300},
   ª = {300,250},       ° = {300,300},        º = {300,250},
   ⁰ = {300,200},
   ¹ = {300,150},       ² = {350,200},        ³ = {250,150},
   ⁴ = {350,100},       ⁵ = {300, 50},        ⁶ = {400,100},
   ⁷ = {400, 50},       ⁸ = {250, 50},        ⁹ = {300, 50},
   ₀ = {300,300},
   ₁ = {300,350},       ₂ = {300,150},        ₃ = {250,250},
   ₄ = {400,200},       ₅ = {300,100},        ₆ = {450,200},
   ₇ = {450,150},       ₈ = {400,250},        ₉ = {400,200},
   ± = {150,100},                             ÷ = {300,300},
   þ = { 50,  },
   †             = {250,200},  ‡              = {250,200},
   ₊ = {300,450},       ₋ = {300,450},
   ⁺ = {300,450},       ⁻ = {300,450},
   - = {300,500},       - = {300,500},        − = {100,300},
   – = {125,305},       — = {200,300},        — = {125,150},
   • = {125,200}

   }

 \SetProtrusion
   [ name     = palatino-sc,
     load     = palatino-default ]
   { encoding = {EU1,EU2,TU},
     family   = {PalatinoLinotype},
     shape    = sc }
   {

   a = {50,50},
   æ = {50,  },
   b = { 0, 0},
   d = { 0, 0},
   f = { 0, 0},
   g = { 0, 0},
   j = {50,  },
   l = {  ,50},
   o = { 0, 0},
   p = { 0, 0},
   q = { 0,  },
   r = {  , 0},
   t = {50,50},
   y = {50,50},
   fl = { 0,50},
   ffl = { 0,50},
   � = { 0,50},
   � = { 0,50}

   }
 ⟨/PalatinoLinotype⟩
```

17 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```
⟨*test⟩
\documentclass{article}

%% Here you can specify the font you want to test, using
%% the commands \fontfamily, \fontseries and \fontshape.
%% Make sure to end all lines with a comment character!
\newcommand*\TestFont{%
  \fontfamily{ppl}%
%%  \fontseries{b}%
%%  \fontshape{it}% sc, sl
}

\usepackage{ifthen}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[verbose,expansion=alltext,stretch=50]{microtype}

\pagestyle{empty}
\setlength{\parindent}{0pt}
\newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
\newcommand*\testprotrusion[2][]{%
  \ifthenelse{\equal{#1}{r}}{}{#2}%
  lorem ipsum dolor sit amet,
    \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
    \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
  you know the rest%
  \ifthenelse{\equal{#1}{l}}{}{#2}%
  \linebreak
  {\fontencoding{\encodingdefault}%
  \fontseries{\seriesdefault}%
  \fontshape{\shapedefault}%
  \selectfont
  Here is the beginning of a line, \dotfill and here is its end}\linebreak
}
\newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
\def\stripprefix#1>{}
\newcount\charcount
\begin{document}

\microtypesetup{expansion=false}

{\centering The font in this document is called by:\\
 \texttt{\showTestFont}\par}\bigskip

\TestFont\selectfont
 This line intentionally left empty\linebreak
%% A -- Z
\charcount=65
\loop
  \testprotrusion{\char\charcount}
  \advance\charcount 1
  \ifnum\charcount < 91 \repeat
%% a -- z
\charcount=97
\loop
  \testprotrusion{\char\charcount}
  \advance\charcount 1
  \ifnum\charcount < 123 \repeat
%% 0 -- 9
\charcount=48
\loop
```

```
  \testprotrusion{\char\charcount}
  \advance\charcount 1
  \ifnum\charcount < 58 \repeat
%%
 \testprotrusion[r]{,}
 \testprotrusion[r]{.}
 \testprotrusion[r]{;}
 \testprotrusion[r]{:}
 \testprotrusion[r]{?}
 \testprotrusion[r]{!}
 \testprotrusion[l]{\textexclamdown}
 \testprotrusion[l]{\textquestiondown}
 \testprotrusion[r]{)}
 \testprotrusion[l]{(}
 \testprotrusion{/}
 \testprotrusion{\char`\\}
 \testprotrusion{-}
 \testprotrusion{\textendash}
 \testprotrusion{\textemdash}
 \testprotrusion{\textquoteleft}
 \testprotrusion{\textquoteright}
 \testprotrusion{\textquotedblleft}
 \testprotrusion{\textquotedblright}
 \testprotrusion{\quotesinglbase}
 \testprotrusion{\quotedblbase}
 \testprotrusion{\guilsinglleft}
 \testprotrusion{\guilsinglright}
 \testprotrusion{\guillemotleft}
 \testprotrusion{\guillemotright}

\newpage
The following displays the current font stretched by 5\%,
normal, and shrunk by 5\%:

\bigskip
\newlength{\MTln}
\newcommand*\teststring
  {ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
\settowidth{\MTln}{\teststring}
\microtypesetup{expansion=true}

\parbox{1.05\MTln}{\teststring\linebreak\\
                   \teststring}\par\bigskip
\parbox{0.95\MTln}{\teststring}

\end{document}
⟨/test⟩
```

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net.

A The title logo

This is `microtype-logo.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a `dtx` file
- `\input` it in the preamble: it then provides the command `\printlogo`, which will do just that

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

*⟨*logo⟩*

Here's how the logo on the title page was created.[30] It has nothing to do with `microtype`, actually, but uses `fontinst`. It is based on an experiment I posted to the `de.comp.text.tex` newsgroup.[31] It will show:

- the character
- the TeX box
- the bounding box
- kerns

A.1 Macros

To run this file, TeX needs to find the `afm` file (either in the `TEXINPUTS` path, or in the current working directory).

First input `fontinst`.

```
\input fontinst.sty
```

`bbox.sty` is an addition to `fontinst`, which makes dimensions of the bounding boxes available (and was written by Hàn Thế Thành, by the way). These dimensions are specified in the `afm` file, but not used by TeX, which is why `fontinst` will discard them otherwise.

```
\input bbox.sty
```

`\tempdim` Allocate some dimen registers.

```
\newdimen\tempdim
```

`\fboxrulei` Frame width of the box as TeX sees it.

```
\newdimen\fboxrulei
\fboxrulei=0.1pt
```

`\fboxruleii` Frame width of the bounding box.

```
\newdimen\fboxruleii
\fboxruleii=0.1pt
```

`\kernboxheight` Height of the box indicating the kern.

```
\newdimen\kernboxheight
\kernboxheight=5pt
```

`\scaletoem` An auxiliary macro. Return a dimension relative to the em-width of the font. Requires e-TeX.

```
\setcommand\scaletoem#1{\dimexpr #1 sp*\fontdimen6\font/1000\relax}
```

`\showlogo` A `fontinst` incantation whose sole purpose is to produce the logo. Its argument is a string (letters only).

```
\fontinstcc
\def\showlogo#1{%
```

Some fonts do not specify the `\fontdimen` 6 (width of an em) in the `afm` file. In this case, use the font size, which is correct in most cases.

```
   \ifdim\fontdimen6\font = 0pt
      \typeout{***~Warning:~no~fontdimen~6~specified~***^^J%
               ***~setting~it~to~\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi~***}
      \fontdimen6\font=\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi\relax
   \fi
  \installfonts
    \input_metrics{}{\logofont,\metrics\printbbs{#1}\relax}
```

---

[30] Note that the `logo` module will not be created when installing `microtype`. Instead, the source file `microtype-logo.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the `pdftk` tool.

[31] Message ID: `42aa3687$0$24366$9b4e6d93@newsread2.arcor-online.net`

```
  \endinstallfonts
}
\normalcc
```

Layers.

```
\makeatletter
\def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
\ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
\ifx\mt@order  \@undefined\let\mt@order  \@empty\fi
\xdef\mt@order{\mt@order[(Logo)}
\let\mtl@resources\@empty
\def\mtl@register#1{%
  \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
  \expandafter\xdef\csname mtl@#1\endcsname{\the\pdflastobj\space 0 R }
  \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
  \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
  \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
\mtl@register{canvas}
\mtl@register{characters}
\mtl@register{bounding-boxes}
\mtl@register{TeX-boxes}
\xdef\mt@order{\mt@order]}
\global\let\mtl@objects\mt@objects
\def\togglelayer#1#2{%
  \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
    user{/Subtype/Link
         /BS << /Type/Border/W 0 >> /H/O
         /A << /S/SetOCGState
               /State[/Toggle \csname mtl@#1\endcsname] >>
  }#2\pdfendlink
}
```

`\printbbs` Preparation.

```
\setcommand\printbbs#1{%
  \setbox0\hbox{#1}%
  \leavevmode
  \kern-\fboxrulei
```

The canvas in the natural width of the text minus protrusion, in color `bgcolor`.

```
  \mtl@layer{canvas}{%
    \getboundarychars#1\relax
    \tempdim=\dimexpr\wd0 - (\scaletoem{\lpcode\font\firstchar}+
                             \scaletoem{\rpcode\font\lastchar})\relax
    \kern\dimexpr\scaletoem{\lpcode\font\firstchar}\relax
    \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
          \hrule width  \tempdim
                 height \dimexpr\dp0+\ht0+0.15em\relax}%
    \kern-\tempdim
```

The baseline, in color `blcolor`.

```
    \vbox{\color{blcolor}%
          \hrule width  \tempdim
                 height \fboxrulei}%
  }%
  \kern-\dimexpr\wd0 -\scaletoem{\rpcode\font\lastchar}\relax
```

The string.

```
  \printbbss #1\relax\relax
}
```

`\getboundarychars` Get first . . . .

```
\def\getboundarychars#1#2\relax{%
   \def\firstchar{`#1}%
   \getlastchar#1#2\relax
}
```

`\getlastchar` . . . and last character.

```
\def\getlastchar#1#2{%
```

```
   \ifx\relax#2\relax
      \def\lastchar{`#1}%
   \else
      \expandafter\getlastchar
   \fi #2%
}
```

`\printbbss` Loop over all characters of the string.

```
\def\printbbss#1#2#3\relax{%
   \ifx\relax#1\relax
   \else
      \ifx\relax#2\relax
         \printbb{#1}{}%
      \else
         \printbb{#1}{#2}%
      \fi
      \expandafter\printbbss
   \fi #2#3\relax
}
```

`\printbb` Record the kern between the current and the following character, then print the character. `\kerning` is a `fontinst` command.

```
\setcommand\printbb#1#2{%
   \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
   \showboxes{#1}%
```

This could be another application.

```
%       \quad
%     w: \the\scaletoem{\width{#1}},
%     bb: \the\scaletoem{\bbleft{#1}}/%
%         \the\scaletoem{\bbright{#1}},
%         \the\scaletoem{\number\numexpr\width{#1}-\bbright{#1}\relax}
%     h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
}
```

`\showboxes` Print the boxes for char ⟨*#1*⟩. This won't work if ⟨*#1*⟩ isn't also the PostScript name of the glyph (e.g., 'comma' ≠ ',').

```
\setcommand\showboxes#1{%
  \leavevmode
  \color{texcolor}%
```

We have to record the width of the glyph.

```
  \setbox0\hbox{{\color{textcolor}#1}}%
  \global\tempdim=\wd0\relax
  \kern-\fboxrulei
```

1. *The TEX box:* Print a frame in color `texcolor`. This frame shows the glyph as TEX sees it.

```
    \mtl@layer{TeX-boxes}{%
      \hbox{%
        \lower\dimexpr \dp0 + \fboxrulei\relax
        \hbox{%
          \vbox{%
            \hrule height\fboxrulei
            \hbox{%
              \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
              \phantom{\unhcopy0}%
              \vrule width\fboxrulei
            }%
            \hrule height\fboxrulei}}}%
    }%
```

2. *The character:* Now we step back and print the actual glyph. We hold it back until now, so that it will be printed *on top* of its box.

```
    \kern-\wd0
    \mtl@layer{characters}{\hbox{\box0}}%
```

Step back by the amount that the character's bounding box differs from the TEX box on the left side.

```
    \kern\dimexpr\scaletoem{\bbleft{#1}}-\tempdim-\fboxruleii\relax
```

3. *The bounding box:* will be printed in color `bbcolor`.

```
      \mtl@layer{bounding-boxes}{%
        {\color{bbcolor}%
        \hbox{%
          \lower\dimexpr-\scaletoem{\bbbottom{#1}}+\fboxruleii\relax
          \hbox{%
            \vbox{%
              \hrule height\fboxruleii
              \hbox to \dimexpr\scaletoem{\numexpr
                           \bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{%
                \vrule height \dimexpr\scaletoem{\numexpr
                                  \bbtop{#1}-\bbbottom{#1}\relax}%
                       width\fboxruleii
                \hfill
                \vrule width\fboxruleii}%
              \hrule height\fboxruleii}}}%
        }%
        \kern-\dimexpr\fboxruleii+\fboxrulei\relax
      }%
```

4. *The kern:* We also print a small box in color `kerncolor` indicating the kerning between the current and the next character; filled for negative kerns, empty for positive kerns.

```
      \kern\scaletoem{\numexpr\width{#1}-\bbright{#1}\relax}%
      \mtl@layer{TeX-boxes}{%
        {\ifnum\thekern<0
           \color{kerncolor}%
           \kern\scaletoem{\thekern}%
           \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoem{\thekern}\relax
                                            height \kernboxheight}%
           \kern\scaletoem{\thekern}%
         \else
           \color{texcolor}%
           \ifnum\thekern=0 \else
             \lower\kernboxheight
             \hbox{%
               \vbox{%
    %            \hrule height\fboxrulei
                 \hbox{%
                   \vrule height \kernboxheight width\fboxrulei
                   \kern\dimexpr\scaletoem{\thekern}-2\fboxrulei\relax
                   \vrule width\fboxrulei
                 }%
               \hrule height\fboxrulei}}%
           \fi
         \fi
        }%
      }%
    %  \kern-\fboxrulei
    }
```

```
\newbox\logobox
\def\printlogo{%
  \setbox\logobox=\hbox{\vbox{%
    \MakePercentComment
```

This is the Kepler MM font used in the logo.

```
    \def\logofont{pkpri9e10}
    \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
    \font\thelogofont=\logofont\space at 82pt
```

This would load the italic Palatino font instead.

```
%\def\logofont{pplri}
%\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
%\edef\logofont{\logofont8r}
%\font\thelogofont=\logofont\space at 78pt
```

Load the font.

```
    \thelogofont
```

Protrusion values (overdone for didactic reasons).

```
    \lpcode\font`M=96
    \rpcode\font`e=46
```

Now we can generate the logo.

```
    \pdfliteral direct{/SXS gs}%
    \showlogo{Microtype}%
%    \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
%    \kern5pt\\[3\baselineskip]
%  \long\def\@makefntext##1{%
%    \leftskip 0pt
%    \parindent 0pt
%    \everypar{\parindent 0pt}%
%    \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
%  \footnotetext[1]{This graphic display on a
%    \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
%    their \togglelayer{bounding-boxes}{bounding boxes}
%    and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
  }}%
  \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
  \immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}%
  \immediate\pdfxform
            attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
            resources {/Properties <<\mtl@resources>>
                       /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
            \logobox
%  \vskip-2.5\baselineskip
%  \leavevmode
%  \togglelayer{characters}{%
%    \pdfrefxform\pdflastxform
%  }%
   \pdfannot\logodimens{%
       /Subtype/Widget /FT/Btn /T(Logo)
       %/F 4 % why did I say this?
       /AP << /N \the\pdflastxform\space 0 R >>
       /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
              /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
              /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
              /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
           >> }%
  \vspace{3\baselineskip}
}
```

Our font.

```
\pdfmapline{+pkpmmri8r10 KeplMM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
```

Define colours (`thered` and `thegreen` are copied from `microtype.dtx`).

```
\def\mtdefinecolors{
 \definecolor{thered}{rgb}{0.65,0.04,0.07}
 \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
 \colorlet{texcolor}{thegreen!50}  % TeX boxes
 \colorlet{kerncolor}{texcolor}    % negative kerns
 \colorlet{bbcolor}{thered!50}     % bounding box
 \colorlet{bgcolor}{black!8}       % canvas
 \colorlet{blcolor}{black!50}      % baseline
 \colorlet{textcolor}{black!40}    % text
}
```

Use with `microtype.dtx`

```
\ifx\documentclass\@twoclasseserror
  \usepackage[xcdraw]{xcolor}
  \mtdefinecolors
\else
```

A.2 Document

Now we can start the document.

```
\documentclass[10pt,a4paper]{ltxdoc}
\providecommand\MakePercentComment{\relax}
\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
```

Re-use the preamble from `microtype.dtx`.

```
\usepackage{microtype-doc}
\usepackage{attachfile}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
\makeatother
\begin{document}
```

You are currently reading this.

```
\DocInput{microtype-logo.dtx}
\newpage
  And here it is:
\vfill
\begin{center}
  \printlogo \null
\end{center}
\vfill
\expandafter\enddocument
\fi
```

That's it.

```
⟨/logo⟩
```

B The letterspacing illustration

This is `microtype-lssample.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a `dtx` file
- `\input` it in the preamble: it then provides the commands
  - `\lssample`: prints the letterspacing illustration
  - `\anchorarrow`: anchors an arrow for layer ⟨*#1*⟩
  - `\showarrow`: toggles layer ⟨*#1*⟩ or ⟨*#2*⟩, and prints ⟨*#2*⟩

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

```
\ifx\lssample\undefined
⟨*lssample⟩
```

Upon popular request, here's how I've created the letterspacing illustration.[32]

B.1 Macros

Rule width and image height and depth.

```
\makeatletter
\newdimen\lsamount
\newdimen\lsrule
\lsrule=0.2pt
\def\lsheight{8pt}
\def\lsdepth{12pt}
```

---

[32] Note that the `lssample` module will not be created when installing `microtype`. Instead, the source file `microtype-lssample.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the `pdftk` tool.

Our font (Adobe Caslon).

```
\def\lsfont{\fontfamily{paca}\selectfont}
```

Loop over all letters in ⟨*#2*⟩, letterspacing them by ⟨*#1*⟩.

```
\def\dols#1#2{\lsamount=#1\relax \dolss#2\enddols}
\def\dolss#1#2\enddols{%
  \ifx\empty#2\empty\divide\lsamount 2\fi
  \ls{#1}%
  \ifx\empty#2\empty\else \dolss#2\enddols \fi
}
```

One `tikz` picture for each letter.

```
\def\ls#1{%
  \begin{tikzpicture}[remember picture,line width=\lsrule]
    \tikzstyle{every node}=[inner sep=0pt]
```

The bounding box.

```
    \mts@layer{stuff}{%
      \node[draw=thegrey,
            fill=theshade,
            outer sep=\lsrule,
            anchor=base,
            font=\lsfont]{\phantom{#1}};
    }
```

The letter.

```
    \node[anchor=base,font=\lsfont](#1){#1};
```

Two auxiliary coordinates.

```
    \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
    \path (#1.base east)  ++(-.5\lsrule,-\lsdepth)  coordinate (#1R);
    \mts@layer{stuff}{%
```

Now draw the normal character width,

```
      \draw[color=thered!75,
            fill=thered!30,
            outer sep=\lsrule]
            (#1L) rectangle (#1R);
      \ifdim\lsamount>0pt
        \path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
        \path (#1R) ++(\lsamount+\lsrule,+\lsdepth) coordinate (#1E);
```

and the letter space.

```
        \draw[color=thered,
              fill=thered!50,
              outer sep=\lsrule]
              (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
      \fi
    }
  \end{tikzpicture}%
  \ignorespaces
}
```

Draw the interword space.

```
\def\lssp#1#2#3#4{%
  \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
    \mts@layer{stuff}{%
      \tikzstyle{every draw}=[anchor=bottom]
      \coordinate(#1space) at (#2/2,\lsdepth/2);
      \coordinate(#1stretch) at (#2+#3/2,+0pt);
      \coordinate(#1shrink) at (#2-#4/2,+0pt);
      \draw[color=thegreen,fill=thegreen!50,use as bounding box]
            (0,0) rectangle ++(+#2,+\lsdepth);
      \draw[color=thegreen,fill=thegreen!30]
            (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
      \draw[color=thegreen,fill=thegreen!50]
            (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
      \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
```

```
            (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
      \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
            (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
    }%
  \end{tikzpicture}%
  \ignorespaces
}
```

Layers.

```
\def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
\def\mtsx@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
\ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
\ifx\mt@order  \@undefined\let\mt@order  \@empty\fi
\xdef\mt@order{\mt@order[(Sheep)}
\let\mts@resources\@empty
\def\mts@register#1{%
  \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
  \expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj\space 0 R }
  \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
  \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
  \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
\mts@register{stuff}
\mts@register{tracking}
\mts@register{ispace}
\mts@register{ospace}
\mts@register{istretch}
\mts@register{ishrink}
\mts@register{ostretch}
\mts@register{oshrink}
\mts@register{okern}
\mts@register{ligature}
\mts@register{_compatibility}
\xdef\mt@order{\mt@order]}
```

Anchor point for the arrow in the code.

```
\newcommand\anchorarrow[1]{%
  \tikz[remember picture,overlay]\node(#1_c){};}
```

Add an arrow from code to image.

```
\newcommand\add@arrow[5][left]{%
  \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
    \mtsx@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
}
```

Toggle layer.

```
\def\toggle@layer#1#2#3{%
  \pdfstartlink
    user{/Subtype/Link
         /BS << /Type/Border/W 0 >> /H/O
%         /BS << /Type/Border/W 1 /S/D /D[4 1] >>
%         /C[0.7 0.7 0.7] /H/O
         /Contents(Click to Toggle!)
         /A << /S/SetOCGState
               /State[/Toggle \csname mts@#1\endcsname] >> }%
  \rlap{#2}%
  {\fboxsep=0pt \fboxrule=0pt
   \mtsx@layer{stuff}{%
     \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
   \mtsx@layer{#1}{%
     \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}%
  }%
  \pdfendlink
}
\newcommand\showarrow[2][]{%
  \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
  \toggle@layer{\@tempa}{{\itshape #2}}}
```

The environment for our illustration.

```
\def\ls@sample#1{{%
  \parskip  4pt \parindent 0pt
  \par
  \vskip4pt
  {\leftskip 15pt
   \mt@pseudo@marg{\color{theblue}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.\strut}
   \mt@layer{_compatibility}{%
      \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
        \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
      \mt@pseudo@marg{\color{thered}%
        If you had a \acronym{PDF} viewer that understands
        \acronym{PDF}\,{\smaller1.5}, you could hide the arrows selectively.}}
   \vskip-\mt@unvdimen}%
  \vskip-4pt
  \setlength\fboxsep{4pt}%
  \leavevmode
  \pdfstartlink
    user{/Subtype/Link
         /BS << /Type/Border/W 0 >> /H/O
         /A << /S/SetOCGState
               /State[/Toggle \mts@stuff] >> }%
    \fcolorbox{theframe}{theshade}%
      {\fontsize{34}{38}\selectfont #1}%
  \pdfendlink
  \par\medskip
  }%
  \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
}
```

Now define the illustration to be used in the document.

```
\def\lssample{%
  \ls@sample{%
    \dols{0pt}{Stop}
      \lssp{o}{0.45em}{0.25em}{0.15em}
    \dols{0.16em}{{st}ealing}\hskip-\dimexpr 0.08em+\lsrule\relax
      \lssp{i}{13.82pt}{4.65pt}{2.08pt}
    \dols{0.16em}{sheep}
    \dols{0pt}{!}
  }%
```

Don’t forget to add the arrows.

```
  \vspace{-\baselineskip}
  \add@arrow{red}      {tracking}{lsamount_c.east}{a_ls}
  \add@arrow{red}      {okern}   {okernend_c.east}{p_ls}
  \add@arrow{green}    {ospace}  {ospace_c.east}  {ospace}
  \add@arrow{green}    {ispace}  {ispace_c.center}{ispace}
  \add@arrow{green!75} {istretch}{istretch_c.east}{istretch.north}
  \add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
  \add@arrow{green!75} {ostretch}{ostretch_c.east}{ostretch.north}
  \add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
  \add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}
}
\fi
```

This is for use with `microtype.dtx`

```
\ifx\documentclass\@twoclasseserror
  \usepackage{tikz}
\else
```

B.2 Document

```
\documentclass[10pt,a4paper]{ltxdoc}
\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
```

Re-use the preamble from `microtype.dtx`.

```
\usepackage{microtype-doc}
\usepackage{attachfile}
\usepackage{tikz}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects]
                             /D << /Order [\mt@order] /BaseState/OFF >> >> }
\makeatother
\begin{document}
```

You are currently reading this.

```
\DocInput{microtype-lssample.dtx}
```

Now show what we are able to do.

```
\noindent
 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):
\begin{verbatim}
\SetTracking
  [ no ligatures  = {"\anchorarrow{nolig}"f},
    spacing       = {60"\anchorarrow{ispace}"0*,"%
                      "-1"\anchorarrow{istretch}"00*, "\anchorarrow{ishrink}"},
    outer spacing = {4"\anchorarrow{ospace}"50,"%
                      "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
    outer kerning = {"\anchorarrow{okernbegin}"*,"%
                      \anchorarrow{okernend}"*} ]
  { encoding = * }
  { 1"\anchorarrow{lsamount}"60 }
\end{verbatim}
 and then write:
\begin{verbatim}
Stop \textls{stealing sheep}!
\end{verbatim}
 this is the (typographically dubious) outcome:

 \lssample

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

\expandafter\enddocument
\fi
⟨/lssample⟩
```

C Change history

2004/09/11 Version 1.0

General: Initial version . . . 1

2004/09/21 Version 1.1

General: configuration file names in lowercase (suggested by *Harald Harders*) . . . 87
  remove 8-bit characters from the configuration files (suggested by *Harald Harders*) . . . 146
Protrusion: add factors for some more characters 153
  settings for Adobe Minion (contributed by *Harald Harders*) . . . 154
`\DeclareCharacterInheritance`: new command: possibility to specify character inheritance . . . 119
`\MT@declare@sets`: remove spaces around set name 105
`\MT@find@file`: fix: also check whether the file for the base font family has already been loaded . . 87
`\MT@get@basefamily`: only remove suffixes ‘x’ or ‘j’ 88
`\MT@get@listname@`: don’t check for empty attributes list . . . 89
`\MT@ifempty`: fix: use category code 12 for the percent character (reported by *Tom Kink*) . . . 45
`\MT@is@number`: numbers may also be specified in hexadecimal or octal (suggested by *Harald Harders*) . . . 94
`\MT@pdftex@no`: fix: version check (reported by *Harald Harders*) . . . 40
`\MT@permute`: don’t use sets for empty encoding . . . 121
`\MT@setup@expansion`: issue an error instead of a warning, when pdfTeX version is too old for `autoexpand` . . . 136
`\MT@split@codes`: fix: allow zero and negative values 63
`\MT@use@set`: remove spaces around set name . . . 109

2004/10/03 Version 1.2

Font aliases: declare `cmor` as an alias of `cmr` . . . 143
Font sets: new: `allmath` and `basicmath` . . . 142
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in `TS1` encoding . . . 177
  add settings for Computer Modern Roman math symbols . . . 182
`\MT@familyalias`: define alias font name as an alternative, not as a replacement . . . 59
`\MT@get@basefamily`: also remove ‘w’ (swash capitals) . . . 88
`\MT@get@highlevel`: check whether defaults have changed . . . 106
`\MT@get@inh@list`: fix: set inheritance list `\globally` to `\@empty` . . . 91
`\MT@get@listname@`: alternatively check for alias font name . . . 89
`\MT@get@size`: additional magic to catch some errors 108
  hijack `\set@fontsize` instead of `\@setfontsize` 107
`\MT@loop`: fix: new macro, used instead of `\loop` . . 49
`\MT@maybe@do`: also check for alias font name . . . 60
`\MT@permute@@@@@@`: more sanity checks for `\SetProtrusion` and `\SetExpansion` . . . 122
`\MT@setupfont`: also search for alias font file . . . 57
  fix: call `\@@enc@update` if necessary . . . 57

2004/10/27 Version 1.3

General: fix: specifying `load` option does no longer require to give a `name`, too . . . 116
Font aliases: declare `aer`, `zer` and `hfor` as aliases of `cmr` . . . 143
`\MT@fix@catcode`: check some category codes (compatibility with `german`) . . . 35
`\MT@load@list`: check whether list exists . . . 87

2004/11/12 Version 1.4

General: check for `pdfcprot` . . . 54
  don’t use scratch registers in global definitions . . 91
  use `\pickup@font` instead of `\define@newfont` as the hook for `\MT@setupfont` . . . 99
  use one instead of five counters . . . 50
Protrusion: tweak quote characters for `cmr` variants (`OT1`, `T1`, `lmr`) . . . 158
`\microtypesetup`: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options . . . 130
`\SetExpansion`: fix: specifying extra options does no longer require to give a `name`, too . . . 113

2004/11/17 Version 1.4a

General: new option: `final` . . . 127
`\MT@cfg@catcodes`: fix: reset some more catcodes when reading files (reported by *Michael Hoppe*) 88

2004/11/26

Version 1.4b

General: fix: set catcodes before reading global configuration file (reported by *Christoph Bier*) . . 129
  optimisation: use less `\expandafter`s and `\csname`s 44
Protrusion: harmonise dashes in upshape and italic (`cmr`, `pad`, `ppl`) 153
  slanted like italics 162
`\MT@checklist@family`: fix: don't try alias family name if encoding failed 61
`\MT@get@basefamily`: fix: failed for font names of the form `abczz` (reported by *Georg Verweyen*) 88
`\MT@get@slot`: don't define `\MT@char` globally (save stack problem) 91
`\MT@ifdimen`: don't set `\MT@count` globally (save stack problem) 46
`\MT@setup@PDF`: new message if `\pdfoutput` is changed 133
`\MT@use@set`: don't use undeclared font sets 109

2004/12/15

Version 1.5

General: defaults: `step`: 4 (suggested by *Hàn Thế Thành*) 127
  new option: `selected`, by default false (suggested by *Hàn Thế Thành*) 125
Documentation: add 'Short history' 30
Inheritance: remove `\ss` from `T1` list, add `\DJ` . . . 146
Protrusion: settings for Bitstream Charter 154
`\DeclareMicrotypeAlias`: remove spaces around arguments 110
`\MT@cfg@catcodes`: reset catcode of '=' (compatibility with Turkish `babel`) 88
`\MT@fix@catcode`: reset catcode of '^' (compatibility with `chemsym`) 35
`\MT@get@highlevel`: don't test defaults if called after begin document 106
`\MT@scale@factor`: warning for factors outside limits 65
`\MT@scale@to@em`: don't use `\lpcode` and `\rpcode` for the calculation 64
`\MT@set@ex@codes`: allow non-selected font expansion 69
`\MT@set@pr@codes`: adjust protrusion factors before setting the inheriting characters 62
`\MT@setup@expansion`: defaults: calculate `step` as min(`stretch`,`shrink`)/5 135
  defaults: turn off expansion for DVI output 134
  disable automatic expansion for DVI output . . . 135

2005/01/24

Version 1.6

General: defaults: turn off expansion for old pdfTEX versions 128
  load a font if none is selected 56
  new option: `factor`, by default 1000 127
  restructure `dtx` file 142
  test whether `\pickup@font` has changed 102
  test whether numeric options receive a number 127
  use e-TEX's `\ifcsname` and `\ifdefined` if defined 44
Protrusion: add italic uppercase Greek letters . . . 162
  improve settings for numbers (pointed out by *Peter Muthesius*) 155
  tune CMR math letters (OML encoding) 182
`\MT@get@charwd`: use e-TEX's `\fontcharwd`, if available 64
`\MT@get@inh@list`: correct message if `selected` is false 90
`\MT@set@ex@codes`: introduce `factor` option 69
`\MT@set@pr@codes`: introduce `factor` option 62
`\MT@setup@expansion`: disable automatic expansion for old pdfTEX versions 136
`\MT@use@set`: retain current set if new set is undeclared 109
`\MT@vinfo`: new macro instead of `\ifMT@verbose` . . 36

2005/02/02

Version 1.6a

Documentation: add table of fonts with tailored protrusion settings 21
`\MT@get@slot`: completely redone, hopefully more robust (compatible with `frenchpro`; problem reported by *Bernard Gaulle*) 91
`\MT@pdftex@no`: new macro 39
`\MT@reset@ef@codes`: only reset `\efcode`s for older pdfTEX versions 70

2005/03/23

Version 1.7

General: allow specification of size ranges (suggested by *Andreas Bühmann*) 106
  disallow automatic expansion if pdfTEX too old 118
  fix: remove space after `autoexpand` 118
  new value for `verbose` option: `errors` 127
  shorter command names 51
  warning when running in draft mode 133
Documentation: add hint about compatibility 26
  remove table of match order (now table 4 on page 89) 12
Protrusion: fix: remove `\` from `OT1`, add `\textbackslash` to `T1` encoding 156
`\LoadMicrotypeFile`: new command (suggested by *Andreas Bühmann*) 111
`\Microtype@Hook`: new command for font package authors 129
`\microtypesetup`: fix: warning also when setting to `(no)compatibility` 130
`\MT@begin@catcodes`: also use inside configuration commands 88
`\MT@cfg@catcodes`: reset catcode of ':' (compatibility with `french*` packages) 88

- `\MT@DeclareMicrotypeAlias`: may also be used inside configuration files 111
- `\MT@get@listname@`: use `\@tfor` (*Andreas Bühmann*'s idea) 89
- `\MT@get@slot`: remove backslash hack 91
  - test for `\chardefed` commands 92
  - test whether \⟨*encoding*⟩\⟨...⟩ is defined 91
- `\MT@if@list@exists`: don't define `\MT@#1@c@name` `\globally`, here and elsewhere 90
- `\MT@ifdimen`: comparison with 1 to allow size smaller than 1 (suggested by *Andreas Bühmann*) 46
- `\MT@increment`: use e-TeX's `\numexpr` if available 50
- `\MT@is@composite`: new macro: construct command for composite character; no uncontrolled expansion 97
- `\MT@scale`: new macro: use e-TeX's `\numexpr` if available 51
- `\MT@set@ex@codes`: two versions of this macro 69
- `\MT@split@name`: don't define `\MT@encoding` &c. `\globally` 59
- `\MT@test@ast`: make it simpler 106
- `\MT@try@order`: always check for size, too (suggested by *Andreas Bühmann*) 89
  - fix: also check for //⟨*series*⟩/⟨*shape*⟩// (reported by *Andreas Bühmann*) 89
- `\MT@warn@code@too@large`: new macro: type out maximum protrusion factor 66
- `\MT@warn@err`: new macro: for `verbose=errors` 36
- `\showhyphens`: modify `\showhyphens` 137

2005/06/23

Version 1.8

- General: `\SetProtrusion`: new key: `unit` 117
  - if font substitution has occurred, set up the substitute font, not the selected one 99
  - new option: `config` to load a different main configuration file 128
  - new option: `unit`, by default `character` 128
- Documentation: add example for `factor` option 13
  - add example of how to get rid of a widow (suggested by *Adam Kucharczyk*) 15
  - add hint about error messages 27
- Font aliases: declare `pxr` and `txr` as aliases of `ppl` resp. `ptm` 144
- Font sets: add `U` encoding to `allmath` 142
- Inheritance: remove `\DJ` from `T1` list (it's the same as `\DH`) 146
- Protrusion: add LY1 characters for Times 161
  - settings for AMS math fonts 185
  - verified settings for slanted Computer Modern Roman 170
- `\add@accent`: fix: disable micro-typographic setup inside `\add@accent` (reported by *Stephan Hennig*) 101
- `\DeclareMicrotypeAlias`: warning when overriding an alias font 110
- `\DeclareMicrotypeSetDefault`: new command: set default font set 109
- `\MT@cfg@catcodes`: reset catcodes of the remaining ASCII characters 88
- `\MT@check@rlist`: made recursive 123
- `\MT@curr@list@name`: new macro: current list type and name 98
- `\MT@declare@sets`: warning when redefining a set 105
- `\MT@define@set@key@`: use comma lists instead of token lists 105
- `\MT@find@file`: no longer wrap names in commands 87
- `\MT@fix@fontdimen@six`: new macro: test whether `\fontdimen` 6 is defined 59
- `\MT@get@charwd`: warning for missing (resp. zero-width) characters 64
- `\MT@get@listname@`: made recursive 89
- `\MT@get@slot`: fix: expand active characters 91
  - test whether \⟨*encoding*⟩\⟨...⟩ is defined made more robust 91
- `\MT@get@unit`: new macro: get unit for codes 67
- `\MT@in@rlist`: made recursive 49
- `\MT@is@active`: new macro: translate `inputenc`-defined characters 95
- `\MT@is@letter`: warning for non-ASCII characters 94
- `\MT@ledmac@setup`: character protrusion with `ledmac` 52
- `\MT@map@clist@n`: new macro: used instead of `\@for` 48
- `\MT@map@tlist@n`: new macro: used instead of `\@tfor` 48
- `\MT@old@cmd`: renamed commands from `\..MicroType..` to `\..Microtype..` 36
- `\MT@pdftex@no`: case 5: pdfTeX 1.30 40
- `\MT@permute@@@@@@`: add ranges to the beginning of the lists 122
- `\MT@scale`: fix: remove spaces in ε-TeX variant (reported by *Mark Rossi*) 51
- `\MT@setupfont@hook`: restore `\%` and `\#` when `hyperref` is loaded 54
  - restore `csquotes`'s active characters 54
  - restore percent character if Spanish `babel` is loaded 54
- `\MT@split@codes`: get character width once only 63
- `\MT@use@set`: fix: remove braces in first line 109
- `\MT@xadd`: simplified 47

2005/10/28

Version 1.9

- General: `\DeclareMicrotypeSet`: new key: `font` 108
  - `\SetProtrusion`: value '`relative`' renamed to '`character`' for key `unit` 117
  - allow context-specific font setup 99
  - compatibility with TeX Live hack (reported by *Herbert Voß*) 39
  - disable `microtype` setup inside `hyperref`'s `\pdfstringdef` (reported by *Hàn Thế Thành*) 55
  - fix: use `true` as the default value 124
  - option `unit`: rename value `relative` to `character` 128
- Documentation: add hint about `verbatim` environment 25
  - add remark about Type 1 fonts required for automatic font expansion 8
- Font aliases: declare `qpl` and `qtm` (`qfonts`, TeX Gyre) as aliases of `ppl` resp. `ptm` 144
- Font sets: add `OT4` encoding to text sets 142
  - add `T5` encoding to text sets 142
- Inheritance: add list for `OT4` 148
  - add list for `T5` (requested by *Hàn Thế Thành*) 149

- Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR 157
  - settings for OT4 encoding (Computer Modern Roman, Palatino, Times) 153
  - settings for T5 encoded Computer Modern Roman 153
- `\DisableLigatures`: new command: disable ligatures (requires pdfTeX 1.30) 111
- `\microtypecontext`: new command: change setup context in the document 103
- `\MT@checklist@family`: fix: add two missing `\expandafters` 61
- `\MT@detokenize@c`: fix the ε-TeX version 45
- `\MT@exp@two@n`: new macros: less `\expandafters` 44
- `\MT@get@opt`: new key ‘`preset`’ to set all characters to the specified value before loading the lists 67
- `\MT@is@active`: redone: use `\set@display@protect` 95
- `\MT@is@letter`: using `\catcode` should be more efficient than inspecting the `\meaning` 94
- `\MT@maybe@do`: redone 60
- `\MT@rem@from@clist`: new macro: remove an item from a comma list 48
- `\MT@scale@factor`: generalised 65
- `\MT@setup@expansion`: disable expansion if both `step` and `shrink` are zero 136
  - warning if user requested zero `step` 135
- `\MT@toks`: use instead of `\toks@` 42
- `\SetProtrusion`: (et al.) new key: `font` 112

2005/12/05 Version 1.9a

- General: ‘⟨*file name*⟩/⟨*line number*⟩’ as default list name 116
  - new option: `defersetup`, by default true 126
  - remove superfluous test whether `\pickup@font` has changed 102
- Documentation: add explanation for error message in DVI mode 27
  - add explanation for error message with non-Type 1 fonts 28
- Font aliases: declare `mdbch` (`mathdesign`) as an alias of `bch` 145
- Protrusion: fix: remove ‘_’ from OT1 encoding 157
  - settings for T5 encoded Charter 153
- `\microtypesetup`: inside the preamble, accepts all package options 130
- `\MT@check@font@cx`: optimise context-sensitive setup 102
- `\MT@define@set@key@`: don’t expand variables immediately (requested by *Georg Verweyen*) 105
- `\MT@get@highlevel`: no longer check whether defaults have changed 106
- `\MT@ifdefined@c@T`: new macros: true case only 44
- `\MT@ifint`: use `\pdfmatch` if available 45
- `\MT@ifstreq`: use `\pdfstrcmp` if available 47
- `\MT@in@clist`: fix 48
- `\MT@info@missing@char`: info instead of warning (after *Michael Hoppe* reported that the ‘fl’ ligature is missing in Palatino SC) 65
- `\MT@is@feature`: new macro: check for pdfTeX feature 51
- `\MT@map@clist@n`: following LaTeX3 48
- `\MT@permute@@@@@`: don’t define permutations for unused encodings 121
- `\MT@rem@from@clist`: fix 48
- `\MT@setup@`: defer setup until the end of the preamble 52

2006/01/20 Version 1.9b

- General: compatibility with `listings`: sanitise more catcodes (reported by *Holger Uhr*) 55
  - compatibility with the `extendedchar` option of the `listings` package 55
- Documentation: activate expansion in the distributed PDF 1
  - add samples of micro-typographic features 4
- `\MT@features`: use throughout the package to adjust to beta-ness 51
- `\MT@ifdimen`: use `\pdfmatch` if available 46
- `\MT@warn@code@too@large`: fix calculation with present factor 66

2006/02/02 Version 1.9c

- Documentation: add example of how to increase protrusion of footnote markers (suggested by *Georg Verweyen*) 22
- Protrusion: settings for URW Garamond 154
- `\MT@define@code@key@font`: fix: context was ignored 115
- `\MT@define@code@key@size`: fix: embrace `\MT@tempsize` in `\csname` (bug introduced in v1.9b) 115

2006/05/05 Version 1.9d

- Font sets: `md*` instead of `m` series in basic sets 142
  - add QX encoding to text sets 142
- Inheritance: add list for QX encoding (contributed by *Maciej Eder*) 148
- Protrusion: settings for QX encoding (contributed by *Maciej Eder*) 160
  - settings for Euro symbols (Adobe, ITC, `marvosym`) 193
  - tweak AMS settings 185
- `\DeclareCharacterInheritance`: fix: empty context 119
- `\MT@detokenize@n`: new macro: use `\detokenize` if available 45
- `\MT@get@ex@opt`: fix: evaluate `preset` 71
- `\MT@get@font@dimen`: warning for zero fontdimen 65
- `\MT@get@opt`: optimise: don’t reset when `preset` option is set 67
  - set list name before presetting 67

- `\MT@is@active`: support for Unicode (`inputenc/utf8`) 95
- `\MT@setupfont@hook`: restore `\%` and `\#` when `tex4ht` is loaded (reported by *Peter Dyballa*) 54
- `\SetProtrusion`: (et al.) optimise: unify keys for mandatory argument 112
  - (et al.) split keys of optional and mandatory argument 112

2006/07/28

Version 1.9e

- General: fix: default value for `activate`: `true` 125
- Documentation: add hint about unknown encodings 26
  - include LPPL 246
- Font aliases: declare `zeur` and `zeus` (`eulervm`) as aliases of `eur` resp. `eus` (`euler`) 145
- Inheritance: adapt to `marvosym`'s changed encoding 150
- Protrusion: complete settings for Euler Fraktur and Script fonts 192
  - fix: forgotten comma in `mt-mvs.cfg`; adapt to `marvosym`'s changed encoding 193
  - settings for Euler Roman font 189
- `\DeclareCharacterInheritance`: new key '`inputenc`' to set the input encoding 119
- `\MT@rem@from@clist`: model after `\@removeelement` 48
- `\MT@setup@`: empty `\MT@setup@` after use (compatibility with the `combine` class) 52
- `\pickup@font`: no tracing with `trace` package 101
- `\SetExpansion`: new key: `inputenc` 113
- `\SetProtrusion`: (et al.) new key: `inputenc` 112

2006/09/09

Version 1.9f

- Protrusion: fix: `euler-vm` did not load `euler` settings 190
- `\MT@curr@list@name`: fix: `\MessageBreak` must not be expanded 98
- `\MT@gdef@n`: new macros: global variants 44
- `\MT@get@inh@list`: fix: input encoding must be set after the inheritance list has been parsed 91
- `\MT@glet`: new macro 44
- `\MT@reset@context`: only reset context if it has actually been changed 103
- `\MT@set@inh@list`: fix: forgotten comma in the features list 119
- `\MT@set@named@keys`: new macro: set name first, simplify parsing of optional argument 114
- `\SetProtrusion`: (et al.) set catcodes before parsing optional argument 112

2007/01/14

Version 2.0

- General: compatibility with `listings`: set catcode of backslash to zero (reported by *Steven Bath*) 55
  - compatibility with `soul`: register `\textls` and `\lsstyle` 55
  - new option: `babel`, by default false (language-dependent setup suggested by *Ulrich Dirr*) 125
  - new option: `letterspace`, by default 100 127
  - new package `letterspace`: a stripped-down version, containing the letterspacing commands only 1
  - option '`babel`': fix: switch off French `babel`'s shorthands properly (reported by *Daniel Flipo*) 141
  - option '`babel`': switch off Turkish `babel`'s shorthands 141
  - option '`unit`', `\SetProtrusion`: deprecate value '`relative`' completely 117
- Documentation: add hint about how to increase `font_max` and `font_mem_size` 28
  - add hint about warning when tracking *and* expansion is applied to a font 28
  - add remark about '`draft`' option disabling `microtype` (noted by *Michalis Miatidis*) 9
  - qualify hint about web documents with regard to older pdfTeX versions 25
  - qualify hints about expansion error messages with regard to older pdfTeX versions 27
- Font sets: new: `footnotesize` and `scriptsize` 142
  - new: `smallcaps` 142
- `\DeclareMicrotypeBabelHook`: new command: interaction with `babel` 112
- `\lsstyle`: fix: font switches don't pose a problem anymore 78
  - fix: letterspacing commands may be nested 78
  - new command: letterspacing 78
  - totally redone, using the new `\letterspacefont` 78
- `\MT@declare@sets`: fix: empty size list when redefining set 105
- `\MT@is@symbol`: made even more robust 96
- `\MT@load@inputenc`: sanitise catcodes before loading input encoding (problem with `listings`) 68
- `\MT@pdftex@no`: case 6: pdfTeX 1.40 40
- `\MT@setup@noligatures`: maybe disable `\MT@noligatures` after the preamble 140
- `\MT@split@name`: adjust to possible letterspacing 59
- `\SetExtraKerning`: new command: additional kerning 114
- `\SetExtraSpacing`: new command: adjustment of interword spacing 113
- `\SetTracking`: new command: tracking 113
- `\textls`: new command: letterspacing 83
  - starred version: remove spaces around text 83
- `\tracingmicrotypeinpdf`: new debug method: mark all fonts with PDF annotations 37

2007/01/21

Version 2.1

- General: compatibility with `pinyin`: disable `microtype` in `\py@macron` (reported by *Sven Naumann*) 56
  - fix: `letterspace` package forgot to load `keyval` 41
- `\lslig`: new command: protect ligatures in letterspaced text 79
- `\MT@get@ls@basefont`: redone: use `\pdfmatch` to make it bullet-proof 79
- `\MT@orig@pickupfont`: compatibility with `CJK`: also check for its definition 100
- `\textls`: fix: use `\hmode@bgroup` 83

2007/07/14

Version 2.2

- General: disable `microtype` if `wordcount` is loaded (reported by *Ross Hetherington*) 52
  - new option: `copyfonts` 126
  - simplify key declarations 116
  - use catcode trickery for e-TeX test 38
- Documentation: add hint about error message with pdfTeX 1.40 27
  - add hint about extra TOC leader dot (first discovered by *Morten Høgholm*) 25
  - add overview 5
  - logo transparency and amusement 1
- Font aliases: declare `chr` (`chmath`) as an alias of `bch` (reported by *Geoff Vallis*) 144
  - declare `fp9x`, `fp9j` (FPL Neu) as aliases of `ppl[xj]` 144
- Font sets: default set for tracking: `smallcaps` 143
- Inheritance: remove '-' → '127' 147
- Protrusion: settings for Bitstream Letter Gothic 154
- Spacing: add sample 194
- Tracking: add ligatures that are to be disabled 150
- `\DeclareMicrotypeVariants`: new command 110
- `\DisableLigatures`: new optional argument: disable selected ligatures only 111
- `\lslig`: always defined 79
- `\MT@checklist@font`: fix: construct font name from characteristics 62
- `\MT@copy@font`: optionally work on copies of fonts 57
- `\MT@get@basefamily`: redone, working on font names and suffixes of arbitrary length 88
- `\MT@get@charwd`: subtract letterspacing amount from width 65
- `\MT@get@ls@basefont`: fix again: remember base font in a macro 79
- `\MT@ifdimen`: employ LuaTeX features if available 46
- `\MT@ifint`: employ LuaTeX features if available 45
- `\MT@ifstreq`: employ LuaTeX features if available 47
  - fix: ε-TeX version shouldn't use `\x` and `\y` (found by *Wiebke Petersen*) 47
- `\MT@is@composite`: more robust: expand exactly once 97
- `\MT@is@symbol`: expand once more (for `frenchpro`) 96
- `\MT@lsfont`: use `\font@name`, not `\MT@font` 75
- `\MT@lua`: (basic) support for LuaTeX 41
- `\MT@pdftex@no`: case 7: pdfTeX 1.40.4 40
- `\MT@preset@aux@space`: generalised 68
- `\MT@set@all@pr`: (et al.) allow empty values 63
- `\MT@set@inputenc@`: only load `inputenc` files if necessary 67
- `\MT@set@tr@codes`: disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4) 76
  - possibility to customise interword spacing 76
- `\MT@setup@expansion`: warning if `stretch` or `shrink` aren't multiples of `step` 136
- `\MT@setupfont`: don't call `\@@enc@update` anymore 57
  - only add features that are available with the respective pdfTeX 57
- `\MT@setupfont@hook`: restore percent character if Galician `babel` is loaded 54
- `\MT@the@pr@code@tr`: adjust protrusion of letterspaced fonts 63
- `\MT@tracking`: remember fonts that shouldn't be letterspaced 74
- `\MT@tracking@`: fix: tracking couldn't be re-enabled 74
- `\MT@warn@tracking@DVI`: warning when letterspacing in DVI mode 139
- `\MT@with@babel@and@T`: also inspect class options 52
- `\pickup@font`: `letterspace`: setup inside group 100
- `\SetTracking`: new key '`no ligatures`' to disable ligatures of letterspaced fonts 113
  - new keys '`spacing`' and '`outer spacing`' to adjust interword spacing (suggested by *Steven E. Harris*) 113
  - third argument may be empty 113
- `\textmicrotypecontext`: new command: wrapper around `\microtypecontext` 103

2007/12/23

Version 2.3

- General: disable `\microtypecontext` in `hyperref`'s `\pdfstringdef` 55
  - fix: really switch off Turkish shorthands 141
  - new value for `verbose` option: `silent` (suggested by *Karl Berry*) 127
  - turned some warnings into errors 127
- Documentation: add kerning sample 18
  - add letterspacing illustration 17
- `\do@subst@correction`: remember substitute font for all times (reported by *Stephan Hennig*) 101
- `\lslig`: redone: extract outer kerns from current letterspacing amount 79
- `\microtypecontext`: made robust (reported by *Stephan Hennig*) 103
- `\MT@begin@catcodes`: fix: don't disable `\KV@@sp@def` 88
- `\MT@define@set@key@font`: `font`: single asterisk means normal font 108
- `\MT@in@clist`: don't use `\x` (reported by *Peter Meier*) 48
- `\MT@is@active`: support for extended Unicode (`inputenc/utf8x` resp. `ucs`) – experimental 95
- `\MT@noligatures`: fix: set evaluation didn't work (bug introduced in v2.2) 85
- `\MT@plain`: `letterspace`: support for `eplain`/`miniltx` 38
- `\MT@set@curr@os`: adjusting spaces made more reliable 76

- `\MT@set@tr@codes`: also adjust tracking if protrusion is not enabled, and even for `letterspace` (reported by *Stephan Hennig*) 76
  - possibility to customise outer kerning (suggested by *Stephan Hennig*) 76
- `\MT@SetTracking`: sanity check for value 113
- `\MT@setup@tracking`: enable protrusion when tracking is enabled 138
- `\MT@tr@outer@l`: only change pre outer space if it contains shrink 81

2008/02/29

Version 2.3a

- General: fix test for `soul` under plain TeX 56
- Documentation: add hint about `babel` having to be loaded first 26
  - add table of available and enabled features 7
  - mention `soulutf8` 30
- Protrusion: adjust LMR quotation marks again 159
- `\MT@error@doesnt@work`: error messages if pdfTeX is too old for extensions 139
- `\MT@fix@catcode`: fix catcodes earlier, and also for the `letterspace` package 35
- `\MT@getkey`: fix: `key=val` in class options list 133
- `\MT@set@codes`: generalised 63
- `\MT@setupfont@hook`: restore percent character if Mexican `babel` is loaded 54

2008/06/04

Version 2.3b

- `\MT@exp@gcs`: new macro: reduce save stack size 44
- `\MT@font@copy`: enable font copies also with protrusion contexts (reported by *Nathan Rosenblum*) 58
- `\MT@get@size`: grouping 107
- `\MT@noligatures@`: fix: warning messages for unknown slots 85
- `\MT@orig@pickupfont`: compatibility with `CJKutf8`: also check for its definition 100
- `\MT@requires@latex`: new macro 38
- `\MT@set@tr@codes`: fix: protrusion adjustment only for new fonts (reported by *Wolfram Schaalo*) 76
- `\MT@tr@outer@l`: fix: only in horizontal mode 81
  - make `\spaceskip`-aware (`ragged2e`) 81
- `\MT@tr@outer@r@`: additional test for horizontal mode 82

2008/11/11

Version 2.3c

- General: LuaTeX supported by default 40
- Documentation: add hint about `spacing` being experimental 26
  - add hint about partial incompatibility with `CJK` 27
- Inheritance: add `\textcommabelow[STst]` to `QX` encoding (reported by *Vasile Gaburici*) 148
- `\MT@detokenize@c`: fix: remove last space only (reported by *Ulrich Dirr*) 45
- `\MT@tr@outer@r@`: additional test for horizontal mode (reported by *Sveinung Heggen*) 82

2009/03/27

Version 2.3d

- General: fix `pinyin` compatibility check (reported by *Silas S. Brown*) 56
  - move setup to the very end (for *Colin Rourke*) 141
- `\ifMT@innannot`: use `pdftexcmds` for debugging 37
- `\lsstyle`: disable for LuaTeX 78
  - make invalid in math mode 78
- `\microtypesetup`: select font after setup 130
- `\MT@check@active@set`: warning for missing default sets 129
- `\MT@lua`: update for LuaTeX 0.36 41
- `\MT@set@tr@codes`: allow zero tracking 75
- `\MT@set@tr@zero`: fix: allow switching off tracking (reported by *Ulrich Dirr*) 79
- `\MT@setup@expansion`: default `step: 1` for pdfTeX versions ≥ 1.40 135
- `\MT@tr@outer@r@`: don't use `\x` (reported by *Ulrich Dirr*) 82
  - fix: don't adjust in math mode (reported by *Christoph Bier*) 82
  - fix: don't adjust inside discretionary (reported by *Maverick Woo*) 82
- `\MT@tr@set@okern`: allow empty value for `outer kerning` 84
- `\textls`: make math mode aware 83

2009/11/09

Version 2.3e

- Documentation: suggest to patch `\@verbatim` instead of `\verbatim` 26
- Expansion: settings for `T2A` encoding (contributed by *Karl Karlsson*) 151
- Font sets: `sc*` instead of `sc` in `smallcaps` set 142
  - add `T2A` encoding 142
- Protrusion: settings for `T2A` encoding (contributed by *Karl Karlsson*) 159
- Spacing: settings for `T2A` encoding (contributed by *Karl Karlsson*) 196
- `\MT@fix@fontdimen@six`: fix: gobbling settings with tracking failed (reported by *Leo*) 59
- `\MT@setup@`: make space-unaware (requested by *Marcin Borkowski*) 52
- `\MT@tikz@setup`: compatibility with `tikz` (first reported by *Christian Stark*) 53
- `\MT@tr@outer@r@`: fix: set current kerning and spacing again (found by *Lars Rönnbäck*) 82

2010/01/10
Version 2.4

- General: new file `microtype.lua` containing the `lua` functions (contributed by *Élie Roux*) 43
- Protrusion: settings for `T2A` encoded Minion (contributed by *Karl Karlsson*) 159

2013/03/13
Version 2.5

- General: allow contexts for LuaTeX 116
  - disable '`DVIoutput`' option for XeTeX 126
  - fix: check whether '⟨*file*⟩/⟨*line*⟩' list name already exists (reported by *Till A. Heilmann*) 116
  - letterspacing with LuaTeX 0.62 74
  - new files: `microtype-pdftex.def`, `microtype-xetex.def`, `microtype-luatex.def`, containing engine-specific definitions 38
  - protrusion with XeTeX 40
  - restore `\space` inside `listings` (reported by *Rolf Dieterich*) 55
- Documentation: add hint about LuaTeX compatibility 27
  - add hint about `spacing` and `ragged2e` 26
  - add hint about `dtx` source code 28
  - include `microtype-logo.dtx` and `microtype-lssample.dtx` 218
- Font aliases: declare `lmsy` and `lmm` as aliases of `cmsy` resp. `cmm` (reported by *Jonas Hogstrom*) 143
  - declare `zgmx` etc. (`garamondx`) as aliases of `ugm` 145
  - declare Latin Modern Roman (OpenType version) as alias of `lmr` when `fontspec` is loaded 143
  - declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) 144
- Font sets: add `EU1` and `EU2` encodings 142
- Inheritance: add rudimentary list for `EU1` and `EU2` 149
- Protrusion: add default lists for `EU1` and `EU2` 157
  - improvements to Computer Modern Roman italics (contributed by *Hendrik Vogt*) 162
- Tracking: add `EU2` encoding to default list 150
- `\DeclareCharacterInheritance`: allow more than one encoding 119
- `\DeclareMicrotypeAlias`: ignore spaces 110
- `\ifMT@nofamily`: info if settings are not family-specific (suggested by *Hàn Thế Thành*) 62
- `\LoadMicrotypeFile`: remove all spaces in font name 111
- `\lsstyle`: fix: ensure to set up math fonts (reported by *RazorXsr*) 78
- `\MT@define@code@key@family`: compatibility with `fontspec`: remove its internal counter (reported by *Till A. Heilmann*) 114
- `\MT@define@code@key@font`: scrub `fontspec` feature count (found by *Meho R*) 115
- `\MT@do@font`: adapt for LuaTeX 50
  - adapt for XeTeX 50
- `\MT@get@slot@`: adapt for LuaTeX (requested by *Georg Duffner*) 92
  - adapt for XeTeX 92
- `\MT@if@outer@next`: fix: conflict with `amsmath` (reported by *Scott Pakin*) 82
- `\MT@info@missing@char`: fix error message for XeTeX (reported by *Juan Acevedo*) 65
- `\MT@is@charx`: compatibility with `xunicode` 96
- `\MT@ledmac@setup`: fix to work with XeTeX (reported by *Maïeul Rouquette*) 53
- `\MT@ls@set@ls`: allow formulas in optional argument to `\textls` (fix by *Heiko Oberdiek*) 84
- `\MT@microtypecontext`: fix: ensure to set up math fonts (reported by *RazorXsr*) 103
- `\MT@register@subst@font`: only register substituted font if it isn't registered already (reported by *George Gratzer* and *Josep Maria Font*) 102
- `\MT@register@subst@font@cx`: only register if it isn't registered already 102
- `\MT@scrubfeatures`: compatibility with `fontspec`: remove its internal counter 60
- `\MT@set@all@pr`: fix: remove space (found by *Meho R*) 63
- `\MT@set@pr@codes`: make info about generic settings encoding-specific (reported by *Sebastian Schubert*) 62
- `\MT@setup@spacing`: warning with `ragged2e` (reported by *Steffen Hoffmann*) 138
- `\MT@setupfont`: select font with `fontspec` (found by *Georg Duffner*) 56
- `\MT@setupfont@hook`: restore `\%` and `\#` when `mathastext` is loaded (found by *Seamus Bradley*) 54

2013/05/23
Version 2.5a

- General: use `luatexbase` instead of `luatextra` (contributed by *Élie Roux*) 43
- Documentation: add notes on typesetting the documentation 28
  - include OpenType configuration files 199
- `\MT@afteraftergroup`: fix: get outer kerning and spacing of nested letterspacing right 77
- `\MT@get@slot@`: adapt to `luaotfload` v2.2 (contributed by *Élie Roux*) 93
- `\MT@led@unhbox@line`: simplified 53
- `\MT@ledmac@setup`: support for `eledmac` 53
- `\MT@ls@outer@k`: add marker for tightly nested letterspacing 85
- `\MT@set@tr@codes`: fix: load font for `fontspec` 76
- `\MT@xspace`: fix `outer spacing` problem with `xspace` (reported by *Dave*) 83

2016/05/01
Version 2.6

- General: load `luaotfload` with LuaTeX 43
  - redefine `\MT@setupfont@hook` globally for problem with `tikzposter` (reported by *Sam Mason*) 54
- Documentation: add hint about partial incompatibility with `xeCJK` and `luatexja` 27
  - missing characters printed with Charis SIL 199

  - suggest to use `etoolbox` to patch `\verbatim` 26
- Font sets: add TU encoding (notified by *Will Robertson*) 142
  - add `si` and `scit` to `smallcaps` set (reported by *uli*) 142
  - new: `allmath-nott` and `alltext-nott` (reqested by *Karl Berry*) 142
- Inheritance: add TU encoding 149
- Protrusion: add TU encoding to lists 157
- Tracking: add TU encoding to default list 150
- `\DeclareMicrotypeSet`: ignore spaces 104
- `\DeclareMicrotypeSetDefault`: ignore spaces . . . 109
- `\DeclareMicrotypeVariants`: ignore spaces 110
- `\lsstyle`: fix: ensure to set up math fonts (reported by *kleenstar*) 78
- `\microtypecontext`: allow `activate` shortcut (reported by *Karl Berry*) 103
- `\MT@declare@sets`: fix: undefine lists for redefining 105
- `\MT@do@font`: speed up for LuaTeX 50
- `\MT@engine`: fix test with LuaTeX 0.85 39
- `\MT@get@slot@`: fix: could fail with XeTeX (reported by *Christopher Schramm*) 92
- `\MT@is@xchar`: update for `fontspec`'s TU encoding . 96
- `\MT@ledmac@setup`: support for `reledmac` 53
- `\MT@luatex@no`: update for LuaTeX 0.85 (renamed primitives) 41
- `\MT@noligatures@`: use `luaotfload` function to keep/inhibit ligatures 86
- `\MT@orig@pickupfont`: (in)compatibility with `luatexja`: disable unknown slots warnings (reported by *Max*) 99
  - (in)compatibility with `xeCJK`: disable unknown slots warnings (reported by *HcN*) 99
  - compatibility with `xeCJK`: pretend that `CJK` wasn't loaded 100
- `\MT@set@tr@codes`: use `luaotfload`'s `kernfactor` feature if available 75
- `\MT@xspace`: fix `outer spacing` problem with (not only) `algorithm` (reported by *Henning* and *Ronnie Marksch*) 83
- `\UseMicrotypeSet`: ignore spaces 109

2016/05/14
Version 2.6a

- General: fixes for `letterspace` package with LuaTeX 49
- `\MT@do@font`: fix `lua` function (reported by *Herbert Voß*) 50
- `\MT@ls@fontspec@font`: fix for value of ±1000 . . . 77

2017/07/07
Version 2.7

- General: drop `luatexbase` with recent LaTeX 43
  - warning with `minimal` class 52
- Documentation: mention that additional kerning does not work in math mode (discovered by *'Daniel'*) 18
- Font aliases: declare aliases for `newpx` 144
  - declare aliases for `newtx` 144
  - declare aliases for `tempora` 144
  - declare aliases for `XCharter` 144
  - declare Latin Modern Roman as alias of `lmr` with new LaTeX format (reported by *Ulrike Fischer*) 143
- Protrusion: automatically choose correct names for Charis SIL small caps (reported by *'ltcomdata'*) 211
- `\lsstyle`: fix: prevent infinite loop with `psnfss` and `exscale` packages (reported by *user11126*, solution by *Ulrike Fischer*) 79
- `\MT@check@range@`: don't warn for override if conflicting list is loaded 123
- `\MT@is@composite`: compatibility with LaTeX 2017/01/01 (`\DeclareUnicodeComposite`) (reported by *Ulrike Fischer* and *'jcr'*) 97
- `\MT@ls@fontspec@font`: fix for '`file:`⟨*font*⟩' spec (reported by *Reinhard Kotucha*) 77
- `\MT@permute@@@@@@`: don't warn for override if conflicting list is loaded 122
- `\MT@reset@ef@codes`: only reset `\efcodes` for older LuaTeX versions 70
- `\MT@setup@expansion`: don't disable automatic expansion for DVI output with LuaTeX 135
- `\MT@tikz@setup`: compatibility with `tikz` (again) . . 53
- `\MT@warn@tracking@DVI`: don't warn for letterspacing in DVI mode with LuaTeX 139

2018/01/14
Version 2.7a

- General: disallow non-automatic expansion with LuaTeX 118
- `\MT@auto`: remove 'autoexpand' for LuaTeX 1.0.6 (reported by *Ulrike Fischer*) 135
  - with LuaTeX, non-automatic font expansion is no longer possible (as confirmed by *Hans Hagen*) 135
- `\MT@get@highlevel`: test whether `\...default` is defined 106
- `\MT@get@slot`: expand active characters earlier . . . 91
- `\MT@info@notracking@`: defer 'No tracking' message 60
- `\MT@is@active`: compatibility with `newunicodechar` (reported by *Nils Anders Danielsson*) 95

2019/02/28
Version 2.7b

- General: update lua function `microtype.info` after changes in `luaotfload` (reported by *Moritz Wemheuer* and *Ulrike Fischer*) 43
- Documentation: update hint about non-7-bit characters (notified by *Frank Mittelbach*) 26
- Inheritance: add textquotedblleft ligature to OT4 (reported by *Franz Wexler*) 148
- `\MT@info@missing@char`: fix message for glyphs specified as names in XeTeX (reported by *Paolo Ney*) 65
- `\MT@setupfont`: always select current font with XeTeX and LuaTeX (reported by *Paolo Ney*, solution by *Ulrike Fischer*) 56

2019/10/10

Version 2.7c

General: turn warning into info when overwriting the `keepligature` function (reported by *Andy N*) . . 86
`\MT@is@active`: compatibility with LaTeX 2019/10/01 95
`\MT@is@symbol`: take care of `\remove@tlig` 96
`\showhyphens`: compatibility with LaTeX 2019/10/01 (reported by *Phelype Oleinik* and *Falk Hanisch*) 137

2019/11/18

Version 2.7d

`\MT@copy@font@`: in LuaTeX, don't use the `\copyfont` primitive, but load the font anew (reported by *Paolo Polesana* and *Oliver Kopp*) 58
`\MT@register@subst@font`: remove substitute font from list (reported by *Markus Kohm*) 102
`\MT@register@subst@font@cx`: remove substitute font from lists 102

2020/12/07

Version 2.8

General: `letterspace` works with e-TeX only 38
compatibility with `soul`: patch for font change (reported by *Md Ayquassar*) 56
fix for `luatexbase` 43
Documentation: declare `DVIoutput` option deprecated 9
squash fake news about automatic font expansion with `dvilualatex` 8
Font aliases: declare aliases for `step` and `domitian` (notified by *Daniel Benjamin Miller*) 144
declare aliases for `stix` and `stix2` fonts 144
declare New Computer Modern as an alias of Latin Modern Roman 144
Font sets: default set for expansion: `alltext-nott` 143
default set for spacing: `alltext-nott` 143
`\lsstyle`: fix: enforce math setup, again 79
`\microtypecontext`: fix `activate` shortcut 103
ignore spaces 103
`\MT@do@font`: fix for XeTeX 50
simplify `lua` function 50
`\MT@fix@fontdimen@six`: try to fix zero `\fontdimen` 6 59
`\MT@if@luaotf@font`: use `lua` function 49
`\MT@ifstreq`: use XeTeX's `\stringcmp` 47
`\MT@setup@expansion`: warning when expanding in DVI mode with LuaTeX (reported by *d909*) . . 134
`\MT@tr@set@space@`: simplified 80
`\MT@tr@unit@`: fix: allow unit regardless whether letterspacing is set 78
`\textmicrotypecontext`: ignore spaces 103

D Index

Numbers in upright shape refer to the page where the corresponding entry is described (bold face) resp. occurs. Numbers in italics refer to the code line where the corresponding entry is defined (underlined) resp. used.

E The LATEX Project Public License

LPPL Version 1.3c 2008-05-04

Preamble

The LATEX Project Public License (LPPL) is the primary license under which the LATEX kernel and the base LATEX 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 TEX-related (such as a LATEX package), but it is written in such a way that you can use it even if your work is unrelated to TEX.

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 LATEX3 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 '`cfgguide.tex`' and '`modguide.tex`' in the base LATEX 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 'LaTeX-Format' or in the case of files belonging to the 'LaTeX-format' a program implementing the 'TeX language'.

Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.
5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
    (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.

    (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

    (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.

    (d) You distribute at least one of the following with the Derived Work:

      i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;

      ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
10. (a) A Derived Work may be distributed under a different license provided that license itself honors

    the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.

    (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work 'as is', without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status 'author-maintained' if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is 'author-maintained'.

The Work has the status 'maintained' if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status 'maintained' to 'unmaintained' if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
    - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
    - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a LaTeX 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 LaTeX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing LaTeX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to LaTeX, the discussion in '`modguide.tex`' may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%   https://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status `maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the 'Work' referring to the three files '`pig.dtx`', '`pig.ins`', and '`pig.sty`' (the last being generated from '`pig.dtx`' using '`pig.ins`'), the 'Base Interpreter' referring to any 'LaTeX-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.