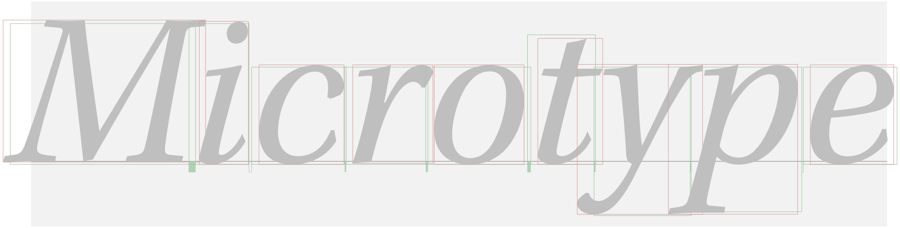

The microtype package

An interface to the micro-typographic extensions of pdfTeX

R Schlicht   w.m.l@gmx.net

v2.0 — 2007/01/21

Abstract

The microtype package provides an interface to the micro-typographic extensions of pdfTeX: 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 ligatures of a font. It allows to apply these features to customisable sets of fonts, and to configure all micro-typographic aspects of the fonts in a straight-forward and flexible way. Settings for various fonts are provided.[1]

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

The alternative package letterspace provides the user commands for letterspacing only, omitting support for all other extensions (see section 7).

[1] Currently, this package provides protrusion settings for Computer Modern Roman, Palatino, Times, URW Garamond, Adobe Garamond and Minion, Bitstream Charter, and the AMS symbols and Euler fonts, for various Euro symbol fonts, as well as some generic settings for unknown fonts. Contributions are very welcome.

Contents

List of Tables

1 Micro-Typography with pdfTeX

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

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

Protrusion	*off*
Expansion	*off*

Both features are enabled throughout this document.

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

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

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

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

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

Robust and hyphenatable *letterspacing* (tracking) has always been extremely difficult to achieve in TeX. Although the `soul` package undertook great efforts in making this possible, it could still fail in certain circumstances; even to adjust the tracking of a font throughout the document remained impossible. Employing pdfTeX's new extension, this doesn't pose a problem any longer. The `microtype` package provides the possibility to change the tracking of customisable sets of fonts, e. g., small capitals. It also introduces two new commands `\textls` and `\lsstyle` for ad-hoc letterspacing, which can be used like the normal text commands.

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 words ends with an ‘r’, the following space should be a tiny bit smaller than that following an, say, ‘m’. You can think of this concept as an extension to TeX’s ‘space factors’. However, while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk –, pdfTeX provides the possibility to modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently have an influence on the interword space. Also, the settings that are shipped with `microtype` are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.8.

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

The possibility, finally, to *disable all ligatures* of a font may be useful for 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 manner. The next chapters will present a survey of all options and customisation possibilities.

2 Invoking the Package

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

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

3 Options

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

3.1 Micro-Typographic Options

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 be enabled, font expansion will only be disabled in circumstances where pdfTeX cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.5). In other words, microtype will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

activate

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

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

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

```
\usepackage{microtype}
```

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

tracking spacing kerning

true, false, ⟨*font set name*⟩ false

The new extensions of tracking, interword spacing, and additional kerning do not have a compatibility level. Therefore, they can only be switched on or off, or they may be activated by passing a set name to the option. By default, neither feature is enabled.

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

3.2 Options for Character Protrusion

`factor` ⟨*integer*⟩ `1000`

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

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

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

3.3 Options for Font Expansion

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

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

Automatic font expansion requires fonts in Type 1 format. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding[2], you should either install the `cm-super` fonts or use the Latin Modern fonts (package `lmodern`).

`stretch` ⟨*integer*⟩ `20`

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

`step` ⟨*integer*⟩ `min(stretch,shrink)/5`

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

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

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

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

```
\usepackage{microtype}
```

`selected` `true, false` `false`

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

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

3.4 Option for Tracking/Letterspacing

`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 thousandth of 1em; admissible values are in the range of −1000 to +1000.

3.5 Miscellaneous Options

`DVIoutput` `true, false` `false`

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

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

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

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

If the `draft` option is passed to the package, *all micro-typographic extensions will be disabled*, 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.

[3] Recent TeX systems are using pdfeTeX as the default engine even for DVI output.

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

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

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

`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`.

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 the keys: `expansion`, `protrusion` and `activate`, which in turn may receive the values `true`, `false`, `compatibility` or `nocompatibility` (but not the name of a font set). Since there is no compatibility level for `tracking`, `spacing` and `kerning`, only the values `true` and `false` are admissible here. Using this command, you could for instance temporarily disable font expansion by saying:

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

4 Selecting Fonts for Micro-Typography

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

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

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

The set of fonts is specified by assigning values to the NFSS font attributes: encoding, family, series, shape and size (cf. LaTeX 2ε font selection). Let's start with an example. This package defines a font set called 'basictext' in the main configuration file as follows:

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

If you now call

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

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

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

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

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

If a value is followed by an asterisk (like 'rm*' and 'sf*' in the above example), it does not designate an NFSS code, but will expand to the document's \⟨*value*⟩default, e. g., \rmdefault. A single asterisk means \⟨*attribute*⟩default, e. g., \encodingdefault, respectively \normalsize for the size axis.

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

Additionally to this declaration scheme, you can add single fonts to a set using the 'font' key, which expects the concatenation of all font 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:

Table 1: Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
`all`	–	–	–	–	–

`alltext`
`(allmath)`
	Text encodings, TS1
	(OML, OMS, U)
		–	–	–	–

`basictext`
`(basicmath)`
	Text encodings
	(OML, OMS)
		`\rm*`,
		`\sf*`
			`\md*`	–	`\normalsize`,
					`\footnotesize`,
					`\small`, `\large`

`smallcaps`	Text encodings	–	–	sc	–

`footnotesize`
`(scriptsize)`
	Text encodings, TS1	–	–	–	`-\small`
					`(-\footnotesize)`

`normalfont`	`\encoding*`	`\family*`	`\series*`	`\shape*`	`\normalsize`

'Text encodings' = OT1, T1, LY1, OT4, QX, T5 '`...*`' = '`...default`'

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

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

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

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

`\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 has been activated in the package options.

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

If the package has been loaded without activating any font sets, the sets declared by this command will be activated. By default, the '`alltext`' font set will be used for character protrusion and additional kerning, the '`basictext`' set for font expansion and interword spacing.

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

5 Micro Fine Tuning

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

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

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

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

5.1 Character Protrusion

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

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

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

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

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

The ⟨*characters*⟩ may be specified either as a single character ('`A`'), as a text symbol command ('`\textquoteleft`'), or as a slot number: three digits for decimal notation, prefixed with `"` for hexadecimal, with `'` for octal (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. You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.6).

[4] For the interested, table 3 on page 67 presents the exact order.

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

Options:

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

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

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

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

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

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

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

`inputenc` Select 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`, with the exception of `uft8x`.

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

5.2 Font Expansion

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

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

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 all expansion factors, to set the input encoding, or to determine the context of the list.

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

As a practical example, suppose you have a paragraph containing a widow that could easily be avoided by shrinking the font a little bit more. You could take advantage of the `stretch` and `shrink` options to allow for more expansion in this particular paragraph. There is one problem that has to be worked around, however: pdfTEX prohibits the use of the same font with different expansion parameters. If you do not want to create a clone of the font setup (this would require duplicating the `tfm`/`vf` files under a new name, and writing new `fd` files and `map` entries), you could exploit a dirty trick and load a minimally larger font for the paragraph in question. E. g., for a document typeset in 10pt:

```
\SetExpansion
   [ stretch = 30,
     shrink  = 60,
     step    =  5 ]
   { encoding = *,
     size = 10.001 }
   { }
\newcommand{\expandpar}[1]{{%[6]
   \fontsize{10.001}{\baselineskip}\selectfont #1\par}}
% ...
\expandpar{This paragraph contains an `unnecessary' widow.}
```

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

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

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

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

If the `selected` option has been passed to the package (cf. section 3.3), and settings for a font don't exist, font expansion will not be applied to this font at all. Should the extraordinary situation arise that you want to employ selected expansion in general but that all characters of a particular font (set) should be expanded or shrunk by the same amount, you would have to declare an empty list for these fonts.

5.3 Tracking

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

An important typographic technique – which was missing in TEX for a long time – is tracking, i. e., the uniform addition or subtraction of letter space to/from all the characters in a font. For example, it is good typographic practice to slightly

[6] 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.

space out pieces of text set in all capitals or small capitals (as in this document), and to increase letter spacing of smaller and decrease that of larger type.[7] With pdfTeX 1.40 and `microtype`, this is now possible, and moreover, easy to achieve. For instance, to increase the letter spacing of all fonts smaller than `\small`, and also space out the small-caps font, you could specify something like the following:

```
\usepackage[tracking=true]{microtype}
\DeclareMicrotypeSet*[tracking]{my}
   { encoding = *,
     size     = -small,
     font     = */*/*/sc/* }
\SetTracking{ encoding = *, shape = sc }{ 50 }
\SetTracking{ encoding = *, size =  footnotesize }{ 70 }
\SetTracking{ encoding = *, size = -footnotesize }{ 100 }
```

In the above example, the small-caps font would be spaced out by 50/1000em (0.025em on each side of each character), fonts in `\footnotesize` would be spaced out by 0.07em, and even smaller type by 0.1em. The amount of letter spacing is given in thousandth of 1em. The interword space will be scaled accordingly.

Letterspaced fonts for which you do not specify an amount will be spaced out by the default of 0.1em (adjustable with the package option `letterspace`, see section 3.5). Suppose your editor wants you to shorten your 1000 pages chef-d'œuvre by 20 pages, you could load `microtype` with (fingers crossed):

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

Options:

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

5.4 Interword Spacing

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

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

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

[7] For 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.

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, however, the settings must contain the two separating commas.

Options:

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

unit Like in \SetProtrusion, you can specify the unit by which the specified numbers are measured. Possible values are: character, a ⟨*dimension*⟩ and, additionally, space. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with these 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 well as the maximum stretch and the maximum shrink amount of the interword space (\fontdimen 3 and 4). As another example, setting all three value to -1000 would completely cancel a space after the respective character.

preset Preset all characters to the specified three values, possibly scaled by a factor, and measured by a given unit (except character).

5.5 Additional Kerning

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

Fine tune the additional 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 additional kerning relates to single characters, that is, whenever that character appears in the text, the specified kerning will be inserted, regardless of which character precedes resp. follows it.

I should not neglect to mention a limitation of this additional kerning: if a word *immediately follows* such a kern (i. e., not separated by a space), hyphenation will be inhibited, unless you insert the breakpoints with \- manually. This restriction of pdfTEX will hopefully be lifted one day.

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 are relative to 1em.

context When it comes to kerning settings, this option is especially useful, since it allows to apply settings depending on the current language (see section 6).

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, with the only exception that exactly one encoding must be specified. The inheritance lists are declared as pairs of ⟨*base character*⟩ = ⟨*list of inheriting characters*⟩. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

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

5.7 Configuration Files

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

If you are embarking on creating new expansion and protrusion settings for a font family, you should put them into a separate file, whose name must be: ‘mt-⟨*font family*⟩.cfg’ (e. g., ‘mt-pad.cfg’), and may contain all commands described in the current section 5. These files will be loaded automatically if you are actually using the respective fonts. If the font name consists of four characters, the package will also try to find the file for the base font family by removing the suffix denoting the sub-family, so that you may put settings for the fonts padx (expert set), padj (oldstyle numerals) and pad (plain) into one and the same file.

This package ships with configuration files for the font families Computer Modern Roman, Palatino, the inescapable Times, URW Garamond, Adobe Garamond and Minion[8], for Bitstream Charter, the AMS symbols and Euler fonts and Euro symbol fonts (Adobe, ITC and marvosym). Table 2 lists them all.

[8] By courtesy of Harald Harders (h.harders@tu-bs.de).

Table 2: Fonts with tailored protrusion settings

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

a Incomplete
b Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)
c Alias: mathdesign/Charter (mdbch)
d Settings inherited from italic shape
e Alias: mathdesign/URW Garamond (mdugm)
f Aliases: pxfonts (pxr), qfonts/QuasiPalatino (qpl)
g Aliases: txfonts (txr), qfonts/QuasiTimes (qtm)
h Alias: eulervm (zeur, zeus)

\DeclareMicrotypeAlias {⟨*font name*⟩} {⟨*alias font*⟩}

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

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

which would make the package, whenever it encounters the font lmr and does not find settings for it, also try the font cmr. In fact, you will find this very line, 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.

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

6 Context-sensitive Setup

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

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

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

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 = footnotes ]
   { 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 `microtype`'s context changed by the footnote marker command. This command differs among the various classes, here are some examples: for the base LaTeX classes, e. g., `article`:

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

For other classes, the command would have to be changed in a similar way.

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 insert

```
\microtypecontext{kerning=}
```

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. 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 has been loaded with the `babel` option (see section 3.5). Currently, `microtype` supports French and Turkish kerning and English spacing (aka. `\nonfrenchspacing`). For unknown languages, all contexts will be reset.

7 Letterspacing revisited

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

`\textls*`
`\lsstyle`

While the `tracking` feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.[10] For such ad-hoc letterspacing, `microtype` introduces two commands that can be used in the same way as LaTeX's text commands: `\textls` expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra space before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by 100/1000em (0.05em on each side); this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option (see section 3.5).

`\lslig` {⟨*ligature*⟩}

One side-effect of letterspacing is that ligatures will be disabled. While this is usually typographically correct, it is not in the case of Fraktur fonts, where the ligatures 'ch', 'ck', 'tz' and 'sz' (ß) should not be broken up. You can protect ligatures (including '`s:`', '`"a`', '`*a`' etc.) with the `\lslig` command:

```
\textfrak{\lsstyle S\lslig{ch}mu\lslig{tz}fle\lslig{ck}} % Schmutzfleck
```

[10] Letterspacing should be used cautiously. Renowned typographers have compared letterspacing lower-case text to stealing sheep (a capital offence in Britain in the 19th century). Unless you know what you are doing, you should probably only letterspace small-capitals or all-capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.

letterspace.sty

These three commands (plus the `letterspace` option) are also available with the `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions. If you find that you don't need `microtype`'s specialties and it is only slowing down the compilation, you should use the `letterspace` package instead (which won't work together with `microtype`; it does not contain anything that `microtype` doesn't).

8 Disabling Ligatures

`\DisableLigatures` {⟨*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* }
```

This command may only be used in the preamble.

9 Hints and Caveats

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

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

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

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

Settings for Cyrillic/Greek/Thai etc. encodings are currently not included. The default sets of fonts for which the micro-typographic features will be enabled (see table 1) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e. g., T2A, LGR etc.), microtype will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

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

You might want to disable protrusion in `verbatim` *environments.* As you know by now, microtype will by default apply character protrusion to all fonts contained in the font set 'alltext'. This also includes the typewriter font. Although it does make sense to protrude the typewriter font if it appears in running text (like, for example, in this manual), this is probably not desirable inside the verbatim environment. However, microtype has no knowledge about the context that a font appears in but will solely decide by examining its attributes. Therefore, you have to take care of disabling protrusion in verbatim environments for yourself (that is, if you don't want to disable protrusion for the typewriter font altogether, by choosing a different font set). While the \microtypesetup command has of course been designed for cases like this, you might find it tiring to repeat it every time if you are using the verbatim environment frequently. The following line, added to the document's preamble, would serve the same purpose:[11]

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

Compatibility. The package should work happily together with all other LaTeX packages (except pdfcprot). However, life isn't perfect, so problems are to be expected. Currently, I am only aware of the following issue:

If you want to use 8-bit characters in the configuration, you have to load the inputenc package first. Unicode input is also supported (when loading inputenc with the utf8 option, however, *not* with the utf8x option resp. the ucs package). If you are using multiple input encodings in your document, 8-bit characters in the settings will not work reliably – you should then specify the inputenc key.

[11] 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.

Possible error messages and how to get rid of them:

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

  This error message will occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running pdfTEX in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your TEX system.

- `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`.

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

  Furthermore, pdfTEX older than 1.40 requires 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 may be increased by setting the respective parameter to a larger value. For web2c-based systems, change the settings in `texmf.cnf`, for MiKTEX, in the file `miktex.ini` (2.4) resp. `pdflatex.ini` (2.5).

- `pdfTeX warning (font expansion): font should be expanded before its first use`

  This warning will occur if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

10 Contributions

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

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

11 Acknowledgments

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

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

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

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulle*, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dyballa*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis* and *Sven Naumann*.

12 References

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

Hàn Thế Thành, *Micro-typographic extensions of 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, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdfTeX user manual*, 1 January 2007. (Available from CTAN at `/systems/pdftex/`; latest version at `http://sarovar.org/projects/pdftex/`)

LaTeX3 Project Team, *LaTeX 2ε font selection*, 27 November 2005. (Available from CTAN at `/macros/latex/doc/fntguide.pdf`)

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

Melchior Franz, *The `soul` package*, 17 November 2003. (Available from CTAN at `/macros/latex/contrib/soul/`)

13 Short History

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

2.1 (2007/01/21)

New command `\lslig` to protect ligatures in letterspaced text (see section 7)

2.0 (2007/01/14)

Support for the new extensions of pdfTeX version 1.40: tracking/letterspacing, adjustment of interword spacing (glue), and additional kerning (new commands `\SetTracking`, `\SetExtraSpacing`, `\SetExtraKerning`; new options '`tracking`', '`spacing`', '`kerning`'; see sections 5.3, 5.4 and 5.5)
New commands `\textls` and `\lsstyle` for letterspacing, new option '`letterspace`' (see sections 3.4 and 7)
New option '`babel`' for automatic micro-typographic adjustment to the selected language (see sections 3.5 and 6)
New font sets: '`smallcaps`', '`footnotesize`', '`scriptsize`' (see section 4 and table 1)
New package '`letterspace`' providing the commands for robust and hyphenatable letterspacing

1.9e (2006/07/28)

New key '`inputenc`' to specify the lists' input encodings (see section 5)
Protrusion settings for Euler math fonts

1.9d (2006/05/05)

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

1.9c (2006/02/02)

Protrusion settings for URW Garamond

1.9a (2005/12/05)

Defer setup until the end of the preamble; consequently, no need to change font defaults before loading `microtype`, or to put it the other way round, `microtype` may now be loaded at any time
Inside the preamble, `\microtypesetup` accepts all package options (see section 3.6)
Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later; see section 8)
New command `\microtypecontext` to change the configuration context; new key '`context`' for the configuration commands (see section 6)
New key '`font`' to add single fonts to the font sets (see section 4)
New key '`preset`' to set all characters to the specified value before loading the lists

Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)
‘DVIoutput’ option will work with TEXLive 2004

1.8 (2005/06/23)

If font substitution has occurred, the settings for the substitute will be used instead of those for the selected font
New command \DeclareMicrotypeSetDefault to declare the default font sets (see section 4)
New option ‘config’ to load a different configuration file (see section 3.5)
New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width (see section 5.1)
Renamed commands from \..MicroType.. to \..Microtype..
Protrusion settings for AMS math fonts
Protrusion settings for Times in LY1 encoding completed
The ‘allmath’ font set also includes U encoding
8-bit characters in the configuration finally work as advertised, even if made active by the csquotes package
When using the ledmac package, character protrusion will work for the first time ever (requires pdfTEX version 1.30 or later)

1.7 (2005/03/23)

Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings (see sections 4 and 5)
Always take font size into account when trying to find protrusion resp. expansion settings for a given font (see section 5)
New command \LoadMicrotypeFile to load a font configuration file manually (see section 5.7)
Hook \Microtype@Hook for font package authors (see section 14.4.3)
New option ‘verbose=errors’ to turn all warnings into errors
Disable expansion inside \showhyphens
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 (see sections 3.2 and 5)
When pdfTEX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
Protrusion settings of digits improved
Use e-TEX extensions, if available

1.5 (2004/12/15)

When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled (see section 3.1)
New option ‘selected’ to enable selected expansion (see sections 3.3 and 5.2); default is: false

New default for expansion option ‘step’: 4 (min(stretch,shrink)/5) (see section 3.3)
Protrusion settings for Bitstream Charter

1.4b (2004/11/26)

\UseMicrotypeSet requires the set to be declared (see section 4)

1.4 (2004/11/12)

Set up fonts independently from LaTeX font loading (therefore, no risk of overlooking fonts anymore, and the package may be loaded at any time)
\microtypesetup now sets the correct level of protrusion (see chapter 3.6)
New option: ‘final’

1.2 (2004/10/03)

New font sets: ‘allmath’ and ‘basicmath’ (see section 4 and table 1)
Protrusion settings for Computer Modern Roman math symbols
Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond
If an alias font name is specified, it will be used as an alternative, not as a replacement (see section 5.7)

1.1 (2004/09/21)

Protrusion settings for Adobe Minion, contributed by Harald Harders
New command: \DeclareCharacterInheritance (see section 5.6)
Characters may also be specified as octal or hexadecimal numbers (see section 5)
Configuration file names in lowercase (see section 5.7)

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).

- debug: Code for additional output in the log file.
  Used for – surprise! – debugging purposes.
- letterspace: The code for the letterspace package (letterspace.sty).

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).
  - 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⟩
⟨*package⟩
```

14.1 Preliminaries

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.

```
\newcommand*\DeclareMicrotypeSet[3][]{}
```

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

This command also has a starred version.

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

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\DeclareMicrotypeBabelHook
```

\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
```

\MT@MT

This is us.

```
\def\MT@MT{microtype}
```

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

Communicate.

```
⟨*!debug⟩
\def\MT@info{\PackageInfo\MT@MT}
\def\MT@info@nl#1{\MT@info{#1\@gobble}}
\let\MT@vinfo\@gobble
\def\MT@warning{\PackageWarning\MT@MT}
\def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
⟨/!debug⟩
⟨/package⟩
⟨*letterspace⟩
\def\MT@warning{\PackageWarning{letterspace}}
```

```
\def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
⟨/letterspace⟩
⟨*package⟩
\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.}}
\def\MT@error{\PackageError\MT@MT}
```

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⟩
\newcount\tracingmicrotype
\tracingmicrotype=\tw@
\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 . . .

```
\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.)

```
\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{\pdfescapestring{#1^^J}}}\fi\fi}
```

\iftracingmicrotypeinpdfall

With \tracingmicrotypeinpdffalse, the PDF output is (hopefully) identical, but some font switches are not 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, a green one marks fonts that have already been 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⟩
```

14.1.2 Requirements

\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 between six cases for pdfTEX:

0: not running pdfTEX

1: pdfTEX (< 0.14f)

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

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

4: + automatic font expansion; default \efcode = 1000 (≥ 1.20)

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

6: + adjustment of interword spacing; extra kerning; \letterspacefont; \pdfmatch[12] (≥ 1.40)

```
⟨/package⟩
\def\MT@pdftex@no{0}
```

A hack circumventing the TEXLive 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 TEXLive 2005.

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

Old packages might have let \pdftexversion to \relax.

[12] This command was actually introduced in 1.30, but was still buggy then.

```
\ifx\pdftexversion\@undefined \else
  \ifx\pdftexversion\relax \else
⟨debug⟩  \MT@dinfo@nl{0}{running pdftex \the\pdftexversion(\pdftexrevision)}
    \def\MT@pdftex@no{6}
    \ifnum\pdftexversion < 140
      \def\MT@pdftex@no{5}
⟨*package⟩
      \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
⟨/package⟩
    \fi
  \fi
\fi
⟨debug⟩\MT@dinfo@nl{0}{pdftex no.: \number\MT@pdftex@no}
```

If we are not using pdfTEX or in case it is too old, we disable everything and exit here.

```
⟨letterspace⟩\ifnum\MT@pdftex@no<6
⟨package⟩\ifnum\MT@pdftex@no<2
  \AtEndOfPackage{\let\@unprocessedoptions\relax}
  \let\CurrentOption\@empty
  \MT@warning@nl{%
    \ifcase\MT@pdftex@no
      You don't seem to be using pdftex.\MessageBreak
    \else
⟨*package⟩
      You are using a pdftex version older than 0.14f.\MessageBreak
      `\MT@MT' won't work with such antiquated versions.\MessageBreak
      Please install a newer version of pdftex.\MessageBreak
⟨/package⟩
⟨*letterspace⟩
      You are using a pdftex version older than 1.40.\MessageBreak
      Sorry, but robust letterspacing doesn't work with\MessageBreak
      this version. Please upgrade%
⟨/letterspace⟩
    \fi
⟨package⟩    All micro-typographic features will be disabled%
  }
\endinput\fi
⟨*package⟩
```

Still there? Then we can begin:

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

```
\def\MT@catcodes{%
  \catcode`\^7 %
  \@makeother\-%
  \@makeother\=%
  \@makeother\*%
  \@makeother\+%
  \@makeother\,%
  \@makeother\/%
  \@makeother\`%
  \@makeother\'%
  \@makeother\"%
  \@makeother\!%
}
```

\MT@restore@catcodes

Polite as we are, we'll restore them afterwards.

```
\def\MT@restore@catcodes#1{%
  \ifx\relax#1\else
    \noexpand\catcode`\noexpand#1\the\catcode`#1\relax
    \expandafter\MT@restore@catcodes
  \fi
}
\edef\MT@restore@catcodes{\MT@restore@catcodes\^\-\=\*\+\,\/\`\'\"\!\relax}
```

```
\MT@catcodes
\AtEndOfPackage{\MT@restore@catcodes}
```

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

```
⟨/package⟩
\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
```

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

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

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

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
\MT@step@default

Default values for expansion.

```
\def\MT@stretch@default{20 }
\def\MT@shrink@default{20 }
\def\MT@step@default{4 }
```

\MT@letterspace
\MT@letterspace@default

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

```
⟨/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
```

14.1.4 Auxiliary Macros

\MT@requires@etex

For definitions that depend on e-TEX features.

```
\expandafter\let\expandafter\MT@requires@etex
  \ifcase 0%
    \ifx\eTeXversion\@undefined 1\else
      \ifx\eTeXversion\relax    1\else
        \ifcase\eTeXversion     1\fi
      \fi
    \fi\space
    \@firstoftwo
  \else
    \@secondoftwo
  \fi
⟨debug⟩\MT@dinfo@nl{0}{\MT@requires@etex{}{not }running etex}
```

\MT@requires@pdftex

For definitions that depend on a particular pdfTEX version.

```
\def\MT@requires@pdftex#1{%
  \ifnum\MT@pdftex@no<#1
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi
}
```

`\MT@glet` The forgotten primitive.

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

`\MT@def@n` `\MT@gdef@n` This is `\@namedef` and global.

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

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

```
⟨/package⟩
\def\MT@edef@n#1{\expandafter\edef\csname #1\endcsname}
⟨*package⟩
\def\MT@xdef@n#1{\expandafter\xdef\csname #1\endcsname}
```

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

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

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

```
\def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}
```

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

```
\def\MT@let@nn#1{\expandafter\MT@let@cn\csname #1\endcsname}
```

`\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.

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

\MT@detokenize@n
\MT@detokenize@c

Translate a macro into a token list. With e-TeX, we can use `\detokenize` (and `\expandafter\string` to get rid of the trailing space). The non-e-TeX version requires some more fiddling (and the `\string` isn't perfect, of course).

```
\MT@requires@etex{
  \def\MT@detokenize@n#1{\detokenize\expandafter{\string#1}}
  \def\MT@detokenize@c#1{\detokenize
    \expandafter\expandafter\expandafter{\expandafter\string#1}}
}{
  \def\MT@detokenize@n#1{\string#1}
  \def\MT@detokenize@c#1{\MT@exp@two@c\zap@space\strip@prefix\meaning#1 \@empty}
}
```

\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
```

\MT@ifint

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

```
⟨package⟩\MT@requires@pdftex6{
  \def\MT@ifint#1{%
    \ifcase\pdfmatch{^-*[0-9]+ *$}{#1}\relax
      \expandafter\@secondoftwo
    \else
      \expandafter\@firstoftwo
    \fi
  }
⟨*package⟩
}{
  \def\MT@ifint#1{%
    \if!\ifnum9<1#1!\else?\fi
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
}
```

\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, it seems.)

```
\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
  }
}{
  \def\MT@ifdimen#1{%
    \setbox\z@=\hbox{%
      \MT@count=1#1\relax
      \ifnum\MT@count=\@ne
        \aftergroup\@secondoftwo
      \else
        \aftergroup\@firstoftwo
      \fi
    }%
  }
}
```

\MT@ifdim Test floating point numbers.

```
\def\MT@ifdim#1#2#3{%
  \ifdim #1\p@ #2 #3\p@
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
```

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

```
\MT@requires@pdftex5{
  \def\MT@ifstreq#1#2{%
    \ifcase\pdfstrcmp{#1}{#2}\relax
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
}{
  \def\MT@ifstreq#1#2{%
    \edef\x{#1}%
    \edef\y{#2}%
    \ifx\x\y
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
}
```

\MT@xadd

Add item to a list.

```
\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
}
```

\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⟩
\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\x##1,#1,##2##3\@nnil{%
    \ifx##2\@empty
      \MT@inlist@false
    \else
```

```
      \MT@inlist@true
    \fi
  }%
  \expandafter\x\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\x##1,#1,##2\x{##1,##2\y}%
  \def\y##1,\y##2\y{\ifx,##1\@empty\else##1\fi}%
  \xdef#2{\MT@exp@two@c\y\x\expandafter,#2,\y,#1,\x}%
}
```

`\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\x{#1}%
  \MT@map@tlist@c#2\MT@in@tlist@
}
\def\MT@in@tlist@#1{%
  \edef\y{#1}%
  \ifx\x\y
    \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.

```
\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*⟩.

```
\def\MT@while@num#1#2#3{%
  \@tempcnta#1\relax
  \MT@loop #3%
    \advance\@tempcnta \@ne
    \ifnum\@tempcnta < #2\MT@repeat
}
```

\MT@do@font

Execute ⟨*#1*⟩ 256 times.

```
\def\MT@do@font{\MT@while@num\z@\@cclvi}
```

\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
\MT@requires@etex{
  \def\MT@increment#1{\edef#1{\number\numexpr #1 + 1\relax}}
}{
  \def\MT@increment#1{%
    \MT@count=#1\relax
    \advance\MT@count \@ne
    \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.

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

\MT@make@string

Set the category code of all characters to 12.

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

\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 `\@tempa`, the type of list to ignore in ⟨*#1*⟩, then comes the action.

```
\def\MT@is@feature#1{%
  \MT@exp@one@n\MT@in@clist\@tempa\MT@features@long
  \ifMT@inlist@
    \expandafter\@firstofone
  \else
    \MT@error{`\@tempa' is not an available micro-typographic\MessageBreak
      feature. Ignoring #1}{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`
- `\showhyphens`

\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⟩
\let\MT@setup@\@empty
```

\MT@addto@setup

We use our private hook to have better control over the timing.

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

It will be executed at the end of the preamble, and emptied (the `combine` class calls it repeatedly).

```
\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
```

\MT@with@package
\MT@with@babel@and

We almost never do anything if a package is not loaded.

```
\def\MT@with@package#1{\@ifpackageloaded{#1}\@firstofone\@gobble}
⟨*package⟩
\def\MT@with@babel@and#1{\@ifpackagewith{babel}{#1}\@firstofone\@gobble}
```

\MT@pdfcprot@error

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

```
\def\MT@pdfcprot@error{%
  \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!}%
  \let\MT@pdfcprot@error\relax
}
\MT@with@package{pdfcprot}\MT@pdfcprot@error
```

Don't load `letterspace`.

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

\MT@ledmac@setup
\MT@led@unhbox@line
\MT@led@kern

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).

```
\MT@requires@pdftex5{
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@ifdefined@c@TF\l@dunhbox@line{%
        \MT@info@nl{Patching ledmac to enable character protrusion}%
        \newdimen\MT@led@kern
        \let\MT@led@unhbox@line\l@dunhbox@line
        \renewcommand*{\l@dunhbox@line}[1]{%
          \ifhbox##1%
            \MT@led@kern=\rightmarginkern##1%
            \kern\leftmarginkern##1%
            \MT@led@unhbox@line##1%
            \kern\MT@led@kern
          \fi
        }%
      }{%
        \MT@warning@nl{%
          Character protrusion in paragraphs with line\MessageBreak
          numbering will only work if you update ledmac}%
      }%
    \fi
  }
}{
  \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 `ledmac' package.\MessageBreak
        Upgrade pdftex to version 1.30 or later}%
    \fi
  }
}
```

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

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

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

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

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

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

Spanish babel modifies \%, storing the original meaning in \percentsign.

```
  \MT@with@babel@and{spanish}{\MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
```

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{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.

```
  \@ifpackageloaded{hyperref}\MT@restore@p@h{%
     \MT@with@package{tex4ht}\MT@restore@p@h}%
}
```

Check again at the end of the preamble.

```
⟨/package⟩
\MT@addto@setup{%
⟨*package⟩
  \MT@with@package{pdfcprot}\MT@pdfcprot@error
  \MT@with@package{ledmac}\MT@ledmac@setup
```

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

```
  \let\MT@setupfont@hook\@empty
  \MT@with@babel@and{spanish}{%
    \g@addto@macro\MT@setupfont@hook{%
      \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
  }
  \MT@with@package{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}%
    }%
  }
⟨/package⟩
  \@ifpackageloaded{hyperref}{%
⟨package⟩    \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
```

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

```
    \pdfstringdefDisableCommands{%
⟨package⟩      \let\pickup@font\MT@orig@pickupfont
      \def\lsstyle{\pdfstringdefWarn\lsstyle}%
      \def\textls#1#{\pdfstringdefWarn\textls}%
    }%
  }{%
⟨*package⟩
    \MT@with@package{tex4ht}{%
      \g@addto@macro\MT@setupfont@hook\MT@restore@p@h
    }%
⟨/package⟩
  }
⟨*package⟩
```

The `listings` package makes numbers and letters active,

```
  \MT@with@package{listings}{%
    \g@addto@macro\MT@normal@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@
```

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.

```
%
⟨/package⟩
  \MT@with@package{soul}{%
    \soulregister\lsstyle 0%
    \soulregister\textls  1%
  }
⟨*package⟩
```

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

```
  \MT@with@package{pinyin}{%
    \let\MT@py@macron\py@macron
    \@ifpackagelater{pinyin}{2006/10/17}{% 4.7.0
      \def\py@macron#1#2{%
        \let\pickup@font\MT@orig@pickupfont
        \MT@py@macron{#1}{#2}%
        \let\pickup@font\MT@pickupfont}%
    }{%
      \def\py@macron#1{%
        \let\pickup@font\MT@orig@pickupfont
        \MT@py@macron{#1}%
        \let\pickup@font\MT@pickupfont}%
    }%
  }
```

```
⟨/package⟩
}
⟨*package⟩
```

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

```
\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`).

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

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

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

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

```
  \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.)

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

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

```
  \MT@tracking
  \MT@check@font
  \ifMT@inlist@
⟨debug⟩\MT@show@pdfannot2%
  \else
    \MT@vinfo{Setting up font `\MT@@font'\on@line}%
```

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

Protrusion has to be set up first, says Thành!

```
    \MT@protrusion
    \MT@expansion
```

Interword spacing and kerning.

```
    \MT@spacing
    \MT@kerning
```

Disable ligatures?

```
    \MT@noligatures
```

Debugging.

```
⟨debug⟩\MT@show@pdfannot1%
```

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

```
    \MT@register@font
  \fi
  \MT@glet\MT@font\@empty
```

```
}
```

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

Split up the font name (⟨*#6*⟩ may be the tracking amount).

```
\def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
  \def\MT@encoding{#1}%
  \def\MT@family  {#2}%
  \def\MT@series  {#3}%
  \def\MT@shape   {#4}%
  \def\MT@size    {#5}%
```

\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}%
}
```

\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)

```
  \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
```

Begin with setting micro-typography to true for this font. The `\MT@checklist@...` tests will set it to false if the property is not in the list. The first non-empty list that does not contain a match will stop us (except for `font`).

```
    \MT@dotrue
    \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@vinfo{... No \@nameuse{MT@abbr@#1}}%
  \fi
}
```

\MT@checklist@

The generic test.

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

Begin a `\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@nl{1}{\@nameuse{MT@abbr@#2}: #1 `\@nameuse{MT@#1}' in list}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#2}: #1 `\@nameuse{MT@#1}' not in list}%
      \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@nl{1}{\@nameuse{MT@abbr@#2}: #1 list empty}}%
}
```

\MT@checklist@font

If the font matches, we skip the rest of the test.

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

There mustn’t be a space after the font name, hence we have to stringify it. There surely is a better way than this silly chain, however, I’m beginning to be haunted by \expandafters in my dreams, so I have to leave it at that.

```
    \expandafter\expandafter\expandafter\MT@exp@one@n
    \expandafter\expandafter\expandafter\MT@in@clist
    \expandafter\expandafter\expandafter{%
      \expandafter\expandafter\expandafter\string
        \expandafter\MT@font\expandafter}%
      \csname MT@#1list@font@\@tempa\endcsname
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: font `\MT@@font' in list}%
      \expandafter\MT@clist@break
    \else
⟨debug⟩\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: font `\MT@@font' not in list}%
      \MT@dofalse
    \fi
  }%
⟨debug⟩  {\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: font list empty}}%
}
```

\MT@checklist@family

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

```
\def\MT@checklist@family#1{%
  \edef\@tempa{\csname MT@#1@setname\endcsname}%
⟨!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@nl{1}{\@nameuse{MT@abbr@#1}: family `\@nameuse{MT@family}' in list}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: family `\@nameuse{MT@family}' not in list}%
      \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@nl{1}{\@nameuse{MT@abbr@#1}: alias `\MT@familyalias' in list}%
          \MT@dotrue
⟨debug⟩\else\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: alias `\MT@familyalias' not in list}%
        \fi
      \fi
    \fi
    \ifMT@do \else
      \expandafter\MT@clist@break
    \fi
  }%
⟨debug⟩  {\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: family list empty}}%
}
```

\MT@checklist@size

Test whether font size is in list of size ranges.

```
\def\MT@checklist@size#1{%
  \edef\@tempa{\csname MT@#1@setname\endcsname}%
⟨!debug⟩  \MT@ifdefined@n@T
⟨debug⟩  \MT@ifdefined@n@TF
      {MT@#1list@size@\@tempa}{%
    \expandafter\MT@in@rlist
        \csname MT@#1list@size@\@tempa\endcsname
    \ifMT@inlist@
⟨debug⟩\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: size `\MT@size' in list}%
      \MT@dotrue
    \else
⟨debug⟩\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: size `\MT@size' not in list}%
      \MT@dofalse
      \expandafter\MT@clist@break
    \fi
  }%
⟨debug⟩  {\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: size list empty}}%
}
```

14.2.1 Protrusion

\MT@protrusion

Set up for protrusion?

```
\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{%
```

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

```
  \MT@if@list@exists{%
    \MT@get@font@dimen@six
    \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@pr@do\@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 \number#1/\number#2}%
  \MT@do@font{%
    \lpcode\MT@font\@tempcnta=#1\relax
    \rpcode\MT@font\@tempcnta=#2\relax
  }%
}
```

\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@gobble@settings
\MT@dimen@six
\MT@get@font@dimen@six

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

```
\def\MT@get@font@dimen@six{%
  \ifnum\fontdimen6\MT@font=\z@
    \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}%
    \expandafter\MT@gobble@settings
  \else
    \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
  \fi
}
\def\MT@gobble@settings#1\@tempc,\relax,{}
```

\MT@pr@do

Split up the values and set `\lpcode` and `\rpcode`.

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

\MT@pr@split

The `keyval` package would remove spaces here, which we needn't do since `\SetProtrusion` ignores spaces in the protrusion list anyway.

```
\def\MT@pr@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \MT@get@char@unit
      \MT@pr@split@val#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=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
  }%
  \def\@tempb{#2}%
  \MT@ifempty\@tempb\relax{%
    \MT@scale@to@em
    \rpcode\MT@font\MT@char=\@tempcntb
⟨debug⟩\MT@dinfo@nl{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char: [#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 @}{%
      \expandafter\MT@map@tlist@c
        \csname MT@inh@\MT@pr@inh@name @\MT@char @\endcsname
        \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 1em of the font.

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

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

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

```
    \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.

```
  \MT@requires@etex{
    \def\MT@get@charwd{%
      \MT@count=\fontcharwd\MT@font\MT@char\relax
      \ifnum\MT@count=\z@ \MT@info@missing@char \fi
    }
  }{
    \def\MT@get@charwd{%
      \setbox\z@=\hbox{\MT@font \char\MT@char}%
      \MT@count=\wd\z@\relax
      \ifnum\MT@count=\z@ \MT@info@missing@char \fi
    }
  }
```

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}
}
```

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

```
\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 \number#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.

```
\MT@requires@etex{
  \def\MT@info@missing@char{%
    \MT@info@nl{Character `\the\MT@toks' \iffontchar\MT@font\MT@char
      has a width of 0pt \else is missing \fi in font\MessageBreak
      `\MT@@font'. Ignoring protrusion settings\MessageBreak
      for this character}%
  }
}{
  \def\MT@info@missing@char{%
    \MT@info@nl{%
      Character `\the\MT@toks' has a width of 0pt\MessageBreak
      (it's probably missing) 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
    \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @max\endcsname
  \else
    \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
      \expandafter\MT@warn@code@too@large\csname MT@\MT@feat @min\endcsname
    \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`, which is being dealt with in `\MT@get@ex@opt`).

```
\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}%
    \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@empty
      \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
                      relative to character widths}%
    \else
      \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\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
  \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\@empty
    \let\MT@get@char@unit\MT@get@charwd
  \else
    \expandafter\ifx\csname MT@\MT@feat @unit@\endcsname\m@ne
      \let\MT@get@space@unit\MT@get@font@dimen
    \else
      \expandafter\MT@get@unit\csname MT@\MT@feat @unit@\endcsname
    \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}%
```

```
  \MT@ifdefined@n@T{%
        MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}{%
    \MT@ifdefined@c@TF\inputencoding{%
```

Set up normal catcodes, since, e. g., listings would otherwise want to typeset the inputenc file when it is loaded inside a listing.

```
      \MT@normal@catcodes
⟨debug⟩\MT@dinfo@nl{1}{input encoding: \@nameuse{%
⟨debug⟩MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}}%
      \inputencoding{\@nameuse{%
        MT@\MT@feat @#1@\csname MT@\MT@feat @#1@name\endcsname @inputenc}}%
    }{%
      \MT@error{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
          \MessageBreak package isn't loaded. Ignoring key}%
        {You must load the `inputenc' package before you can use the `inputenc' key.}%
    }%
  }%
}
```

\MT@set@pr@heirs

Set the inheriting characters.

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

\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
    \let\MT@preset@aux\MT@preset@aux@space
  \fi
  \MT@preset@aux{#1}\@tempa
  \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{%
  \def\@tempb{#1}%
  \MT@get@space@unit\tw@
  \MT@scale@to@em
  \edef#2{\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}%
}
```

14.2.2 Expansion

\MT@expansion

Set up for expansion?

```
\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
    \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@ex@do\@tempc,\relax,%
    \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
  }\relax
}
```

\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.

```
\newif\ifMT@nonselected
```

```
\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
  \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
  \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@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 versions prior to 1.20.

```
\MT@requires@pdftex4{
  \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@do

There's only one number per character.

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

\MT@ex@split

```
\def\MT@ex@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \@tempcntb=#2\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: [#2]}%
```

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 @}{%
          \expandafter\MT@map@tlist@c
            \csname MT@inh@\MT@ex@inh@name @\MT@char @\endcsname
            \MT@set@ex@heirs
        }%
      }%
    \fi
  \fi
}
```

\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@}%
  \def\@tempa{autoexpand}%
  \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
  \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}%
  \MT@dinfo@nl{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
⟨/debug⟩
}
```

\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
}
```

14.2.3 Interword Spacing (Glue)

\MT@spacing Adjustment of interword spacing?

```
\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@font@dimen@six
    \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@sp@do\@tempc,\relax,%
  }\MT@reset@sp@codes
}
```

\MT@sp@do

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

\MT@sp@split

```
\def\MT@sp@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \MT@get@char@unit
      \MT@sp@split@val#2\relax
    \fi
  \fi
}
```

\MT@split@val

If unit=space, \MT@get@space@unit will be defined to fetch the corresponding fontdimen (2 for the first, 3 for the second and 4 for the third argument).

```
\def\MT@sp@split@val#1,#2,#3\relax{%
  \def\@tempb{#1}%
  \MT@ifempty\@tempb\relax{%
    \MT@get@space@unit\tw@
    \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@unit\thr@@
    \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 @}{%
      \expandafter\MT@map@tlist@c
        \csname MT@inh@\MT@sp@inh@name @\MT@char @\endcsname
        \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}%
  \MT@dinfo@nl{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
                                           \number\stbscode\MT@font\MT@char/%
                                           \number\shbscode\MT@font\MT@char}%
⟨/debug⟩
}
```

\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 \number#1/\number#2/\number#3}%
  \MT@do@font{%
    \knbscode\MT@font\@tempcnta=#1\relax
    \stbscode\MT@font\@tempcnta=#2\relax
    \shbscode\MT@font\@tempcnta=#3\relax
  }%
}
\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@preset@aux@factor{#1}\@tempa
    \MT@preset@aux@factor{#2}\@tempc
    \MT@preset@aux@factor{#3}\@tempb
  \else
    \MT@preset@aux@space{#1}\@tempa
    \def\@tempb{#2}%
    \MT@get@space@unit\thr@@
    \MT@scale@to@em
    \edef\@tempc{\number\@tempcntb}%
    \def\@tempb{#3}%
    \MT@get@space@unit4%
    \MT@scale@to@em
    \edef\@tempb{\number\@tempcntb}%
  \fi
  \MT@set@all@sp\@tempa\@tempc\@tempb
}
```

Only for sufficiently new versions of pdfTEX.

```
}{
  \let\MT@spacing\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@font@dimen@six
    \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@kn@do\@tempc,\relax,%
  }\MT@reset@kn@codes
}
```

\MT@kn@do

```
\def\MT@kn@do#1,{%
  \ifx\relax#1\@empty \else
```

```
    \MT@kn@split #1==\relax
    \expandafter\MT@kn@do
  \fi
}
```

\MT@kn@split

```
\def\MT@kn@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \MT@get@char@unit
      \MT@kn@split@val#2\relax
    \fi
  \fi
}
```

\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@unit\tw@
    \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@unit\tw@
    \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 @}{%
      \expandafter\MT@map@tlist@c
        \csname MT@inh@\MT@kn@inh@name @\MT@char @\endcsname
        \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}%
  \MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char/%
                                 \number\knaccode\MT@font\MT@char}%
⟨/debug⟩
}
```

\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 \number#1/\number#2}%
  \MT@do@font{%
    \knbccode\MT@font\@tempcnta=#1\relax
    \knaccode\MT@font\@tempcnta=#2\relax
  }%
}
\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
    \let\MT@preset@aux\MT@preset@aux@space
  \fi
  \MT@preset@aux{#1}\@tempa
  \MT@preset@aux{#2}\@tempb
  \MT@set@all@kn\@tempa\@tempb
}
```

```
}{
  \let\MT@kerning\relax
}
⟨/package⟩
```

14.2.5 Tracking

\lsstyle

This only works with pdfTEX 1.40.

```
⟨package⟩\MT@requires@pdftex6{
  \DeclareRobustCommand\lsstyle{%
⟨*package⟩
    \MT@trackingtrue
    \def\MT@tr@setname{all}%
⟨/package⟩
    \let\MT@tracking\MT@tracking@
```

Enable the (protected) `\lslig` command.

```
    \MT@let@nc{lslig }\MT@lslig
    \selectfont
```

Possibly remove the first kern (for `\textls*`). Use scaling to avoid a ‘Dimension too large’.

```
    \MT@lskern=\dimexpr\MT@ifdefined@c@TF\MT@letterspace@
                         \MT@letterspace@\MT@letterspace sp
                       * \fontdimen6\MT@lsfont/2000\relax
    \MT@ls@adjust
  }
```

\MT@tracking

\MT@tracking@

We do not check whether we’ve already seen the font!

```
  \def\MT@tracking@{%
⟨package⟩     \ifMT@tracking\MT@maybe@do{tr}\fi
⟨letterspace⟩     \MT@set@tr@codes
  }
⟨package⟩  \let\MT@tracking\MT@tracking@
⟨letterspace⟩  \let\MT@tracking\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.

```
  \def\MT@set@tr@codes{%
    \MT@ifdefined@c@TF\MT@letterspace@\relax{%
      \let\MT@letterspace@\MT@letterspace
⟨*package⟩
```

```
   \MT@if@list@exists{%
    \MT@set@listname
    \MT@let@cn\MT@letterspace@{MT@tr@c@\MT@tr@c@name}%
```

Different unit?

```
    \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\else
      \MT@get@font@dimen@six
      \MT@let@cn\@tempb{MT@tr@c@\MT@tr@c@name}%
      \MT@get@unit\MT@tr@unit@
      \let\MT@tr@factor@\@m
      \MT@scale@to@em
      \edef\MT@letterspace@{\number\@tempcntb}%
     \fi
    }%
   }\relax
⟨/package⟩
  }%
  \ifnum\MT@letterspace@=\z@\else
```

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

```
   \xdef\MT@lsfont{%
⟨package⟩      \expandafter\csname\expandafter\string\MT@font
⟨letterspace⟩    \csname\curr@fontshape\f@size
                   /\number\MT@letterspace@ ls\endcsname}%
   \expandafter\ifx\MT@lsfont\relax
```

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

```
    \MT@get@ls@basefont
⟨debug⟩\MT@dinfo@nl{1}{... new letterspacing instance}%
    \global\expandafter\letterspacefont\MT@lsfont\font@name \MT@letterspace@
```

Scale interword spacing (`\fontdimen` 2).

```
    \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@\relax sp
                  * \fontdimen2\MT@lsfont/1000\relax
   \fi
⟨*package⟩
   \MT@vinfo{Tracking font `\MT@@font' by \number\MT@letterspace@
           \space\on@line}%
   \let\MT@font\MT@lsfont
   \aftergroup
⟨/package⟩
   \MT@set@lsfont
  \fi
 }
```

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

```
 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}
```

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

```
 \DeclareRobustCommand\lslig{\@firstofone}
 \def\MT@lslig#1{%
  \MT@get@ls@basefont
  \kern\MT@lskern{\font@name #1}\kern\MT@lskern
 }
```

\MT@get@ls@basefont

pdfTEX cannot letterspace fonts that are already letterspaced. Therefore, we have to extract the base font, if we find that `\font@name` contains a '+' or a '-' (with `\pdfmatch` bullet-proof even if the font name contains one of these letters).

```
  \def\MT@get@ls@basefont{%
    \ifcase\pdfmatch{^select font (.*)[+-][0-9]*ls( at .*)?$}
                    {\expandafter\meaning\font@name}\relax
    \else
⟨debug⟩\MT@dinfo@nl{1}{... fixing base font}%
      \global\expandafter\font\font@name
        \expandafter\strip@prefix\pdflastmatch1
        \expandafter\strip@prefix\pdflastmatch2\relax
    \fi
  }
⟨*package⟩
```

For older pdfTEX versions, let it relax.

```
}{
  \let\MT@tracking\relax
  \DeclareRobustCommand\lsstyle{%
    \MT@warning{Letterspacing only works with pdftex version 1.40\MessageBreak
      or newer. You might want to use the `soul' package\MessageBreak
      instead}%
    \MT@glet\lsstyle\relax
  }
}
⟨/package⟩
```

\textls

This command may be used like the other text commands.

```
\DeclareRobustCommand\textls{%
  \hmode@bgroup
  \@ifstar{\let\MT@ls@adjust\MT@ls@adjust@\MT@textls}%
          {\let\MT@ls@adjust\relax        \MT@textls}%
}
```

\MT@textls
\MT@letterspace@

The optional argument may be used to change the letterspacing factor.

```
\newcommand\MT@textls[2][]{%
  \MT@ifempty{#1}%
    {\let\MT@letterspace@\@undefined}%
    {\KV@@sp@def\MT@letterspace@{#1}%
     \MT@ls@too@large\MT@letterspace@}%
  \lsstyle #2%
  \MT@ls@adjust
  \egroup
}
```

\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@lskern

This dimen is used for the starred version of `\textls` and for `\lslig`.

```
\newdimen\MT@lskern
```

\MT@ls@adjust \MT@ls@adjust@ Remove extra space for the starred version.

```
\let\MT@ls@adjust\relax
\def\MT@ls@adjust@{\kern-\MT@lskern}
⟨*package⟩
```

14.2.6 Disabling Ligatures

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

```
\MT@requires@pdftex5{
  \def\MT@noligatures{%
    \MT@dotrue
    \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
      \pdfnoligatures\MT@font
      \MT@vinfo{... Disabling ligatures}%
    \fi
  }
}{
  \let\MT@noligatures\relax
}
```

14.2.7 Loading the Configuration

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

```
\def\MT@load@list#1{%
  \edef\@tempa{#1}%
  \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
  \MT@ifstreq\@tempa\@tempb{%
    \MT@warning{\@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\csname MT@\MT@feat @do\expandafter\endcsname\@tempc,\relax,%
      }{%
        \MT@warning{\@nameuse{MT@abbr@\MT@feat} list `\@tempb' undefined.
                    Cannot load\MessageBreak 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,}%
      }{%
        \expandafter\MT@get@basefamily#1\relax\relax\relax
        \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
        \ifMT@inlist@\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@normal@catcodes We have to make sure that all characters have the correct category code. Especially, new lines and spaces should be ignored, since files might be loaded in the middle of the document. This is basically \nfss@catcodes (from the LaTeX kernel). I've added: & (in tabulars), !, ?, ;, : (french), ,, $, _, ~, and = (Turkish babel).

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

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

```
\def\MT@normal@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 the configuration commands \Set..., and \DeclareCharacterInheritance, so that the catcodes are also harmless when these commands are used outside the configuration files.

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

Table 3: Order for matching font attributes

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Encoding	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Family	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-
Series	●	●	●	●	-	-	-	-	●	●	●	●	-	-	-	-
Shape	●	●	-	-	●	●	-	-	●	●	-	-	●	●	-	-
Size	●	-	●	-	●	-	●	-	●	-	●	-	●	-	●	-

```
  \MT@normal@catcodes
```

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

```
  \let\KV@@sp@def\def
}
```

\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 for expert or old style number font set or for swash capitals (x, j or w). We mustn't simply remove the last letter, as this would make for instance cms out of cmss *and* cmsy (OK, cmex will still become cme . . .).

```
\def\MT@get@basefamily#1#2#3#4\relax{%
  \ifx#2\relax \def\@tempa{#1}\else
    \ifx#3\relax \def\@tempa{#1#2}\else
      \def\@tempa{#1#2#3}%
      \ifx\relax#4\relax \else
        \MT@ifstreq{#4}{\string x}\relax{%
          \MT@ifstreq{#4}{\string j}\relax{%
            \MT@ifstreq{#4}{\string w}\relax{%
              \def\@tempa{#1#2#3#4}}}}\fi\fi\fi
}
```

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

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

```
\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 3 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{%
  \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
    \expandafter\MT@in@rlist\csname MT@\@tempb @\@tempa @sizes\endcsname
    \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
                    `\MT@listname'}%
⟨/debug⟩
    \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,%
      \global\MT@let@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

\MT@char@

The character is in \@tempa, we want its slot number in \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}%
```

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@
```

- 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
```

- 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@
  \ifnum\MT@char < \z@
    \MT@warn@unknown
  \else
```

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

```
    \ifMT@norest \else
      \MT@warn@rest
      \let\MT@char\m@ne
    \fi
  \fi
  \escapechar\m@ne
}
```

`\ifMT@norest` `\MT@testrest` Switch and test whether all of the string has been used up.

```
\newif\ifMT@norest
\def\MT@testrest#1#2{\MT@ifstreq{#1}{#2}\relax\MT@norestfalse}
```

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

```
\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@ > 127 \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@ > \@cclv
        \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) are also supported (however, not ucs, aka.

inputenc/utf8x).

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

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}%
      \edef\x{%
        \def\noexpand\@tempa{\@tempa}%
```

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

```
        \MT@toks={\the\MT@toks\space(= \@tempa)}%
      }%
    \expandafter\endgroup\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@
```

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

```
    \expandafter\MT@is@letter\MT@char\relax\relax
  \fi
}
```

\MT@is@char
\MT@charstring

A helper macro that inspects the \meaning of its argument.

```
\begingroup
  \catcode`\/=0

  /MT@map@tlist@n{/\CHAR}/@makeother
  /lowercase{%
    /def/x{%
      /def/MT@charstring{\CHAR"}%
      /def/MT@is@char##1\CHAR"##2##3##4/relax{%
        /ifx/relax##1/relax
          /if##3\/relax
            /edef/MT@char@{/number"##2}%
            /MT@testrest/MT@charstring{##3##4}%
          /else
            /edef/MT@char@{/number"##2##3}%
            /MT@testrest/MT@charstring{##4}%
          /fi
⟨debug⟩  /MT@dinfo@nl{3}{> `/the/MT@toks' is a \char (/MT@char@)}%
        /fi
      }%
    }%
  }
/expandafter/endgroup/x
```

\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 expands to a letter if it has been defined by \DeclareTextComposite. This should be robust, finally, especially, since we also \detokenize the input instead of only \stringifying it. Thus, we will die gracefully even on wrong Unicode input without utf8.

```
    \edef\MT@char{\expandafter\csname\expandafter
                    \string\csname\MT@encoding\endcsname
                    \MT@detokenize@n{#1}-%
                    \MT@detokenize@n{#2}%
                  \endcsname}%
    \expandafter\MT@is@letter\MT@char\relax\relax
  \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 \mathcodes and \mathchardefs have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e. g., the minus in cmsy when the euler package is loaded). So, no way to go, unfortunately.)

Some warning messages, for performance reasons separated here.

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

\MT@set@listname

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

For my reference:

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

- \get@external@font is called by \extract@font, by itself, and by the substitution macros

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

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

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

\MT@font Additionally, we hook into \do@subst@correction, which is called if a substitution has taken place, to record the name of the ersatz font. Unfortunately, this will only work for one-level substitutions.

All this is done at the beginning of the document.

```
\let\MT@font\@empty
⟨/package⟩
\MT@addto@setup{%
⟨*package⟩
  \g@addto@macro\do@subst@correction{%
    \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
  }
⟨/package⟩
```

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

```
  \@ifpackageloaded{CJK}{
    \@ifpackagelater{CJK}{2006/10/17}
      {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}} % 4.7.0
      {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}
    \g@addto@macro\MT@orig@pickupfont{%
      {\expandafter\ifx\font@name\relax\define@newfont\fi}%
      {\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}}
  }{
    \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
  }
  \ifx\pickup@font\MT@orig@pickupfont \else
    \MT@warning@nl{%
      Command \string\pickup@font\space is not defined as expected.%
⟨*package⟩
      \MessageBreak Double-check whether micro-typography is indeed\MessageBreak
      applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)%
⟨/package⟩
    }
  \fi
```

Then we append our stuff. Everything is done inside a group. *package

```
  \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{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
⟨/package⟩
  \g@addto@macro\pickup@font{%
⟨letterspace⟩    \MT@tracking
⟨*package⟩
⟨*debug⟩
      \global\MT@inannottrue
      \MT@glet\MT@pdf@annot\@empty
      \MT@addto@annot{(line \number\inputlineno)}%
⟨/debug⟩
      \escapechar\m@ne
```

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.

```
      \ifx\MT@font\@empty
        \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
    \endgroup
⟨/package⟩
  }
⟨*package⟩
```

\MT@pickupfont

Remember the patched command for later.

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

\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. At first, I was going to change \hmode@bgroup only, but that is also used in the commands defined by \DeclareTextFontCommand, i. e., \textit etc.

```
  \let\MT@orig@add@accent\add@accent
  \def\add@accent#1#2{%
    \let\pickup@font\MT@orig@pickupfont
    \MT@orig@add@accent{#1}{#2}%
    \let\pickup@font\MT@pickupfont
  }
⟨/package⟩
}
⟨*package⟩
```

Consequently, we are the last one 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@subst@font Register the substituted font.

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

\MT@register@font Register the current font.

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

14.2.10 Context-sensitive Setup

Here are the variants for context-sensitive setup.

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

```
\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.

```
\def\MT@register@subst@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \expandafter\MT@xadd
      \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
      {\font@name,}%
  }%
}
```

\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{%
    \expandafter\ifx\csname MT@\@nameuse{MT@abbr@##1}\endcsname\relax\else
      \expandafter\MT@xadd
        \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
        {\MT@font,}%
      \def\@tempa{##1}%
      \expandafter\MT@map@tlist@c
        \csname MT@##1@doc@contexts\endcsname
        \MT@rem@from@lists
    \fi
  }%
}
```

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

```
\def\MT@rem@from@lists#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

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

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

```
\def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
\MT@addto@setup{%
  \def\microtypecontext#1{%
    \MT@setup@contexts
    \let\MT@reset@context\relax
    \setkeys{MTC}{#1}%
    \selectfont
    \MT@reset@context
  }%
}
```

\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
    {\global\MT@let@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
}
```

\MT@define@context

```
\def\MT@define@context#1{%
  \define@key{MTC}{#1}[]{%
    \KV@@sp@def\@tempb{#1}%
    \edef\@tempb{\@nameuse{MT@rbba@\@tempb}}%
    \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{@}}{\KV@@sp@def\MT@val{##1}}%
      \expandafter\ifx\csname MT@\@tempb @context\endcsname\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.

```
        \global\MT@let@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
          \expandafter\MT@xadd\csname MT@\@tempb @doc@contexts\endcsname{{\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
  }%
}
```

```
\MT@map@clist@c\MT@features@long{\MT@define@context{#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{%
  \@ifstar
    {\@ifnextchar[\MT@DeclareSetAndUseIt
                  {\MT@DeclareSetAndUseIt[]}}%
    {\@ifnextchar[\MT@DeclareSet
                  {\MT@DeclareSet[]}}%
}
```

\MT@DeclareSet
\MT@DeclareSetAndUseIt

```
\def\MT@DeclareSet[#1]{%
  \MT@DeclareSet@{#1}%
}
\def\MT@DeclareSetAndUseIt[#1]#2#3{%
  \MT@DeclareSet@{#1}{#2}{#3}%
  \UseMicrotypeSet[#1]{#2}%
}
```

\MT@DeclareSet@

```
\def\MT@DeclareSet@#1#2#3{%
  \KV@@sp@def\@tempa{#1}%
  \MT@ifempty\@tempa{%
    \MT@map@clist@c\MT@features{{\MT@declare@sets{##1}{#2}{#3}}}%
  }{%
    \MT@map@clist@c\@tempa{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{set declaration `#2'}{%
```

```
          \MT@exp@one@n\MT@declare@sets
            {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
        }%
      }%
    }}%
  }%
}
```

\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{%
  \KV@@sp@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'}%
    \global\MT@let@nc{MT@#1list@size@\MT@curr@set@name}\@empty
  }%
  \global\MT@let@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@keys

Define the keyval keys for font sets.

```
\def\MT@define@set@keys#1{%
  \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}%
}
```

\MT@define@set@key@

⟨*#1*⟩ = font axis, ⟨*#2*⟩ = feature.

```
\def\MT@define@set@key@#1#2{%
  \define@key{MT@#2@set}{#1}[]{%
    \global\MT@let@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 lead to \rmdefault resp. \bfdefault being expanded/protruded.

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

And 'family = *' will become \familydefault.

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

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

```
  }%
}
```

\MT@test@ast

Test whether last character is an asterisk.

```
\def\MT@test@ast#1*#2\@nil{%
  \def\@tempa{#1}%
  \MT@ifempty{#2}\@gobble\@firstofone
}
```

\MT@font@sets
\MT@fix@font@set

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

```
\let\MT@font@sets\@empty
\def\MT@fix@font@set#1{%
  \xdef#1{#1}%
  \global\MT@make@string#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}{%
      \KV@@sp@def\MT@val{####1}%
      \expandafter\MT@get@range\MT@val--\@nil
      \ifx\MT@val\relax \else
        \expandafter\MT@xadd
          \csname MT@#1list@size@\MT@curr@set@name\endcsname
          {{{\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 is trying to do this for the OpenType version of Adobe's Minion. See `http://developer.berlios.de/projects/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}}%
```

2048pt 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@warning{%
       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, and not \@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{\gdef\MT@val{##2}}%
   \@tempa\@nil
  \endgroup
 \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}[]{%
    \global\MT@let@nc{MT@#1list@font@\MT@curr@set@name}\@empty
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \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@ifempty{#1#2#3#4#5}\relax{%
    \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}%
    \MT@ifempty{#5}{%
      \MT@warn@axis@empty{size}{\string\normalsize}%
      \def\MT@val{*}%
    }{%
      \def\MT@val{#5}%
    }%
    \MT@get@size
    \ifx\MT@val\relax\def\MT@val{0}\fi
    \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
    \let\MT@val\@tempb
  }%
}
```

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

```
\MT@map@clist@c\MT@features{\MT@define@set@keys{#1}}
```

It is also used for \DisableLigatures.

```
\MT@define@set@keys{nl}
```

\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.

```
\renewcommand*\UseMicrotypeSet[2][]{%
  \KV@@sp@def\@tempa{#1}%
  \MT@ifempty\@tempa{%
    \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
  }{%
    \MT@map@clist@c\@tempa{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{activation of set `#2'}{%
          \MT@exp@one@n\MT@use@set
            {\csname MT@rbba@\@tempa\endcsname}{#2}%
        }%
      }%
    }%
  }%
}
```

\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{%
  \KV@@sp@def\@tempa{#2}%
  \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
    \MT@xdef@n{MT@#1@setname}{\@tempa}%
    \MT@info{Using \@nameuse{MT@abbr@#1} set `\@tempa'}%
  }{%
    \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
      \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
    }%
    \MT@warning{%
      The \@nameuse{MT@abbr@#1} set `\@tempa' is undeclared.\MessageBreak
      Using set `\@nameuse{MT@#1@setname}' instead}%
  }%
}
```

\DisableLigatures

This is really simple now: we can re-use the set definitions of `\DeclareMicrotypeSet`; there can only be one set, which we'll call '`no ligatures`'.

```
\MT@requires@pdftex5{
  \renewcommand*\DisableLigatures[1]{%
    \edef\MT@active@features{\MT@active@features,nl}%
    \MT@noligaturestrue
    \MT@declare@sets{nl}{no ligatures}{#1}%
    \gdef\MT@nl@setname{no ligatures}%
  }
}{
```

If pdfTEX is too old, we issue a warning and neutralise the command.

```
  \renewcommand*\DisableLigatures[1]{%
    \MT@warning{Disabling ligatures of a font is only possible\MessageBreak
      with pdftex version 1.30 or newer.\MessageBreak
      Ignoring \string\DisableLigatures}%
    \MT@glet\DisableLigatures\@gobble
  }
}
```

\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.

```
\renewcommand*\DeclareMicrotypeSetDefault[2][]{%
  \KV@@sp@def\@tempa{#1}%
  \MT@ifempty\@tempa{%
    \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
  }{%
    \MT@map@clist@c\@tempa{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{declaration of default set `#2'}{%
          \MT@exp@one@n\MT@set@default@set
            {\csname MT@rbba@\@tempa\endcsname}{#2}%
        }%
      }%
    }}%
  }%
}
```

\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{%
  \KV@@sp@def\@tempa{#2}%
  \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
⟨debug⟩\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `\@tempa'}%
    \MT@xdef@n{MT@default@#1@set}{\@tempa}%
  }{%
    \MT@warning{%
      The \@nameuse{MT@abbr@#1} set `\@tempa' is not declared.\MessageBreak
      Cannot make it the default set. Using set\MessageBreak `all' instead}%
    \MT@xdef@n{MT@default@#1@set}{all}%
  }%
}
```

\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.

```
\renewcommand*\DeclareMicrotypeAlias[2]{%
  \KV@@sp@def\@tempa{#1}%
  \KV@@sp@def\@tempb{#2}%
  \MT@make@string\@tempb
  \MT@ifdefined@n@T{MT@\@tempa @alias}{%
    \MT@warning{Alias font family `\@tempb' will override
      alias `\@nameuse{MT@\@tempa @alias}'\MessageBreak
      for font family `\@tempa'}}%
  \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%
```

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

```
  \MT@ifdefined@c@T\MT@family{%
⟨debug⟩\MT@dinfo{1}{Activating alias font `\@tempb' for `\MT@family'}%
    \MT@glet\MT@familyalias\@tempb
  }%
}
```

\LoadMicrotypeFile

May be used to load a configuration file manually.

```
\def\LoadMicrotypeFile#1{%
  \KV@@sp@def\@tempa{#1}%
  \MT@make@string\@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
}
```

14.3.2 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.

```
\def\DeclareMicrotypeBabelHook#1#2{%
  \MT@map@clist@n{#1}{%
    \KV@@sp@def\@tempa{##1}%
    \MT@gdef@n{MT@babel@\@tempa}{#2}%
  }%
}
```

14.3.3 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).

```
\def\SetProtrusion{%
  \MT@begin@catcodes
  \MT@SetProtrusion
}
```

\MT@SetProtrusion
\MT@pr@c@name
\MT@extra@context
\MT@permutelist

We want the catcodes to be correct even if this is called in the preamble.

```
\newcommand*\MT@SetProtrusion[3][]{%
  \let\MT@extra@context\@empty
```

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
}
```

\SetExpansion

\SetExpansion only differs in that it allows some extra options (stretch, shrink, step, auto).

```
\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}%
      \global\MT@let@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

```
\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
  \MT@gdef@n{MT@tr@c@\MT@tr@c@name}{#3}%
  \MT@end@catcodes
}
```

\SetExtraSpacing

```
\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
}
```

\MT@set@named@keys
\MT@options

We first set the name (if specified), then remove it from the list, and set the remaining keys.

```
\def\MT@set@named@keys#1#2{%
  \def\x##1name=##2,##3\@nil{%
    \setkeys{#1}{name=##2}%
    \def\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@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
        \expandafter\MT@xadd\csname MT@tempsize\endcsname
           {{{\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}%
      \expandafter\MT@get@font@and@size\MT@val/////\@nil
      \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
                      \ifx\MT@extra@context\@empty\else\MessageBreak
                        (context: \MT@extra@context)\fi}%
⟨/debug⟩
      \expandafter\MT@xaddb
        \csname MT@\MT@permutelist @\@tempb\MT@extra@context @sizes\endcsname
        {{{\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@ifempty{#1#2#3#4#5}\relax{%
    \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}%
```

Append an asterisk for the size.

```
    \edef\@tempb{\@tempb*}%
    \MT@ifempty{#5}{%
      \MT@warn@axis@empty{size}{\string\normalsize}%
      \def\MT@val{*}%
    }{%
      \def\MT@val{#5}%
    }%
    \MT@get@size
  }%
}
```

```
\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@keys

The options in the optional first argument.

```
\def\MT@define@opt@keys#1{%
```

Use file name and line number as the list name if the user didn't bother to invent one.

```
  \define@key{MT@#1@c}{name}[]{%
    \MT@ifempty{##1}{%
      \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}}}%
}
```

\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@map@clist@c\MT@features{\MT@define@opt@keys{#1}}
```

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

```
\define@key{MT@pr@c}{unit}[character]{%
  \global\MT@let@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{%
      \global\MT@let@nc{MT@pr@c@\MT@curr@set@name @unit}\@tempa
    }{%
      \MT@warning{`\@tempa' is not a dimension.\MessageBreak
        Ignoring it and setting values relative to\MessageBreak
        character widths}%
    }%
  }%
}
```

Tracking may only be relative to a dimension.

```
\define@key{MT@tr@c}{unit}[1em]{%
  \global\MT@let@nc{MT@tr@c@\MT@curr@set@name @unit}\@empty
  \def\@tempa{#1}%
  \MT@ifdimen\@tempa{%
    \global\MT@let@nc{MT@tr@c@\MT@curr@set@name @unit}\@tempa
  }{%
    \MT@warning{`\@tempa' is not a dimension.\MessageBreak
      Ignoring it and setting values relative to\MessageBreak
      1em}%
    \MT@gdef@n{MT@tr@c@\MT@curr@set@name @unit}{1em}%
  }%
}
```

\MT@define@key@unit

Spacing and kerning codes may additionally be relative to space dimensions.

```
\def\MT@define@key@unit#1{%
  \define@key{MT@#1@c}{unit}[space]{%
    \global\MT@let@nc{MT@#1@c@\MT@curr@set@name @unit}\@empty
    \def\@tempa{##1}%
    \MT@ifstreq\@tempa{character}\relax{%
      \global\MT@let@nc{MT@#1@c@\MT@curr@set@name @unit}\m@ne
      \MT@ifstreq\@tempa{space}\relax{%
        \MT@ifdimen\@tempa{%
          \global\MT@let@nc{MT@#1@c@\MT@curr@set@name @unit}\@tempa
        }{%
          \MT@warning{`\@tempa' is not a dimension.\MessageBreak
            Ignoring it and setting values relative to\MessageBreak
            width of space}%
        }%
      }%
    }%
  }%
}
```

```
\MT@define@key@unit{sp}
\MT@define@key@unit{kn}
```

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

```
\def\MT@define@ex@opt@key#1{%
  \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}%
      }%
    }%
  }%
}
```

```
\MT@define@ex@opt@key{stretch}
\MT@define@ex@opt@key{shrink}
\MT@define@ex@opt@key{step}
\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.

```
    \MT@requires@pdftex4{%
      \MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}%
    }{%
      \MT@warning{pdftex too old for automatic font expansion}%
    }
  \else
    \MT@requires@pdftex4{%
      \global\MT@let@nc{MT@ex@c@\MT@curr@set@name @auto}\@empty
    }\relax
  \fi
}
```

14.3.4 Character Inheritance

\DeclareCharacterInheritance This macro may be used in the configuration files to declare characters that should inherit protrusion resp. expansion values from other characters. Thus, there is no need to define all accented characters (e. g., \`a, \'a, \^a, \~a, \"a, \r{a}, \k{a}, \u{a}), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

\MT@inh@feat The optional argument may be used to restrict the list to some features,
\MT@extra@inputenc and to specify an input encoding.

```
\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.

```
\define@key{MT@inh}{encoding}[]{%
  \def\MT@val{#1}%
  \expandafter\MT@encoding@check\MT@val,\@nil
  \MT@get@highlevel{encoding}%
  \MT@edef@n{MT@tempencoding1}{\MT@val}%
}
```

\MT@encoding@check

But we only allow *one* encoding.

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

For the rest, we can reuse the key setup from the configuration lists (\Set...).

```
\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.

```
\def\MT@inh@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \MT@get@slot
    \ifnum\MT@char > \m@ne
      \let\MT@val\MT@char
      \MT@map@clist@n{#2}{%
        \def\@tempa{##1}%
        \ifx\@tempa\@empty \else
          \MT@get@slot
          \ifnum\MT@char > \m@ne
            \expandafter\MT@xadd
              \csname MT@inh@\MT@listname @\MT@val @\endcsname
              {{\MT@char}}%
          \fi
        \fi
      }%
⟨*debug⟩
      \MT@dinfo@nl{2}{children of #1 (\MT@val):
                      \@nameuse{MT@inh@\MT@listname @\MT@val @}}%
⟨/debug⟩
    \fi
  \fi
}
```

14.3.5 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.

```
\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).

```
  \def\@tempb{////}%
  \ifx\@tempa\@tempb \else
    \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
        }%
        \expandafter\MT@xaddb
          \csname MT@\MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
          \MT@tempsize
⟨*debug⟩
        \MT@dinfo@nl{1}{initialising: use list for font \@tempa,\MessageBreak
                 sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
                                @sizes\endcsname}%
⟨/debug⟩
      }{%
```

Only one list can apply to a given combination.

```
        \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context}{%
          \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
            `\@nameuse{MT@\MT@permutelist @name}' will override list\MessageBreak
            `\@nameuse{MT@\MT@permutelist @\@tempa\MT@extra@context}'
            for font `\@tempa'}%
        }%
⟨*debug⟩
        \MT@dinfo@nl{1}{initialising: use list for font \@tempa
                        \ifx\MT@extra@context\@empty\else\MessageBreak
                          (context: \MT@extra@context)\fi}%
⟨/debug⟩
      }%
      \MT@xdef@n{MT@\MT@permutelist @\@tempa\MT@extra@context}%
          {\csname MT@\MT@permutelist @name\endcsname}%
    \fi
  \fi
}
```

\MT@permute@define

Define the commands.

```
\def\MT@permute@define#1{%
  \expandafter\@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
  \expandafter\MT@map@tlist@c
    \csname MT@\MT@permutelist @\@tempa\MT@extra@context @sizes\endcsname
    \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, ⟨*#2*⟩ and ⟨*#3*⟩ those of the existing range.

```
\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@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@define@option expansion and protrusion may be true, false, compatibility, nocompatibility and/or a ⟨*set name*⟩.

```
\def\MT@define@option#1{%
  \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@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}%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}
```

```
\MT@define@option{protrusion}
\MT@define@option{expansion}
```

activate is a shortcut for protrusion and expansion.

```
\define@key{MT}{activate}[true]{%
  \setkeys{MT}{protrusion={#1}}%
  \setkeys{MT}{expansion={#1}}%
}
```

\MT@define@option@ spacing, kerning and tracking do not have a compatibility level.

```
\def\MT@define@option@#1{%
  \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@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}%
            }%
          }%
        }%
      }%
    }%
  }%
}
```

```
\MT@define@option@{spacing}
\MT@define@option@{kerning}
\MT@define@option@{tracking}
```

\MT@def@bool@opt

The true/false options: draft, final (may be inherited from the class options), auto, selected, babel, DVIoutput, defersetup.

```
\def\MT@def@bool@opt#1#2{%
  \define@key{MT}{#1}[]{%
    \MT@ifempty{##1}%
      {\def\@tempa{true}}%
      {\def\@tempa{##1}}%
    \MT@ifstreq\@tempa{true}\relax{%
      \MT@ifstreq\@tempa{false}\relax{%
        \MT@warning@nl{%
          `##1' is not an admissible value for option\MessageBreak
          `#1'. Assuming `false'}%
        \def\@tempa{false}%
      }%
    }%
    #2%
  }%
}
```

\MT@def@simple@bool@opt

Boolean options that only set the switch.

```
\def\MT@def@simple@bool@opt#1{\MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
```

```
\MT@map@tlist@n{{draft}{auto}{selected}{babel}}\MT@def@simple@bool@opt
```

The DVIoutput option will change \pdfoutput immediately to minimise the risk of confusing other packages.

```
\MT@def@bool@opt{DVIoutput}{%
  \csname if\@tempa\endcsname
    \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
    \pdfoutput\z@
  \else
    \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
    \pdfoutput\@ne
  \fi
}
```

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.

```
\MT@def@bool@opt{defersetup}{%
  \csname if\@tempa\endcsname \else
    \AtEndOfPackage{%
      \MT@setup@
      \let\MT@setup@\@empty
      \let\MT@addto@setup\@firstofone
    }%
  \fi
}
```

final is the opposite to draft.

```
\MT@def@bool@opt{final}{%
  \csname if\@tempa\endcsname
    \MT@draftfalse
  \else
    \MT@drafttrue
  \fi
}
```

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

```
\define@key{MT}{verbose}[]{%
  \let\MT@vinfo\MT@info@nl
  \MT@ifempty{#1}%
    {\def\@tempa{true}}%
    {\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
      \MT@ifstreq\@tempa{false}\relax{%
        \MT@warning@nl{%
          `#1' is not an admissible value for option\MessageBreak
          `verbose'. Assuming `false'}%
      }%
    }%
  }%
}
```

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

```
⟨/package⟩
\def\MT@def@num@opt#1{%
  \define@key{MT}{#1}[]{%
    \MT@ifempty{##1}%
      {\MT@let@cn\@tempa{MT@#1@default}}%
      {\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@warning@nl{%
        Value `##1' for option `#1' is not a number.\MessageBreak
        Using default value of \number\@nameuse{MT@#1@default}}%
    }%
```

```
  }%
}
```

```
⟨package⟩\MT@map@tlist@n{{stretch}{shrink}{step}{letterspace}}\MT@def@num@opt
⟨letterspace⟩\MT@def@num@opt{letterspace}
⟨*package⟩
```

`factor` will define the protrusion factor only.

```
\define@key{MT}{factor}[]{%
  \MT@ifempty{#1}%
    {\let\@tempa\MT@factor@default}%
    {\def\@tempa{#1 }}%
  \MT@ifint\@tempa{%
    \edef\MT@pr@factor{\@tempa}%
  }{\MT@warning@nl{%
      Value `#1' for option `factor' is not a number.\MessageBreak
      Using default value of \number\MT@factor@default}%
  }%
}
```

Unit for protrusion codes.

```
\define@key{MT}{unit}[]{%
  \MT@ifempty{#1}%
    {\def\@tempa{character}}%
    {\KV@@sp@def\@tempa{#1}}%
  \MT@ifstreq\@tempa{character}\relax{%
    \MT@ifdimen\@tempa{%
      \let\MT@pr@unit\@tempa
    }{%
      \MT@warning@nl{`\@tempa' is not a dimension. Ignoring it and\MessageBreak
                     setting values relative to character widths}%
    }%
  }%
}
```

14.4.2 Reading the Configuration File

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as TEX systems are switching to the 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
\ifnum\pdfoutput<\@ne \else
```

Also, we only enable expansion by default if pdfTEX can expand the fonts automatically.

```
  \MT@requires@pdftex4{
    \MT@expansiontrue
    \MT@autotrue
  }\relax
\fi
```

`\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.

```
\define@key{MT}{config}[]{\relax}
\def\MT@get@config#1config=#2,#3\@nil{%
  \MT@ifempty{#2}%
```

```
    {\def\MT@config@file{\MT@MT.cfg}}%
    {\KV@@sp@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}%
}
```

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

```
\MT@map@clist@c\MT@features{%
  \MT@ifdefined@n@TF{MT@default@#1@set}\relax
    {\MT@gdef@n{MT@default@#1@set}{all}}%
}
```

\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}'}%
  }{%
    \global\MT@let@nn{MT@#1@setname}{MT@default@#1@set}%
    \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
  }
}
```

14.4.3 Hook for Other Packages

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

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using \@ifpackageloaded in the font package has not been 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
```

14.4.4 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{\setkeys{MTX}}}
\def\MT@define@optionX#1#2{%
  \define@key{MTX}{#1}[true]{%
    \KV@@sp@def\@tempb{#1}%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
      \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\@tempb{%
            \expandafter\@tempcnta=\csname MT@\@tempb @level\endcsname
            \MT@info{Enabling #1
                    (level \number\csname MT@\@tempb @level\endcsname)}%
          }%
        }{%
          \MT@ifstreq\MT@val{false}{%
            \@tempcnta=\z@
            \MT@info{Disabling #1}%
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@checksetup\@tempb{%
                \@tempcnta=\@ne
                \MT@let@nc{MT@\@tempb @level}\@ne
                \MT@info{Setting #1 to level 1}%
              }%
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@checksetup\@tempb{%
                  \@tempcnta=\tw@
                  \MT@let@nc{MT@\@tempb @level}\tw@
                  \MT@info{Setting #1 to level 2}%
                }%
              }{%
```

```
                \MT@warning{%
                  Value `\MT@val' for key `#1' not recognised.\MessageBreak
                  Use any of `true', `false', `compatibility' or\MessageBreak
                  `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{%
  \expandafter\csname ifMT@\csname MT@abbr@#1\endcsname\endcsname
    \expandafter\@firstofone
  \else
    \MT@warning{%
      You cannot enable \@nameuse{MT@abbr@#1} if it was disabled\MessageBreak
      in the package options,}%
    \expandafter\@gobble
  \fi
}
```

```
\MT@define@optionX{protrusion}\pdfprotrudechars
\MT@define@optionX{expansion}\pdfadjustspacing
```

\MT@define@optionX@

The same for `tracking`, `spacing` and `kerning`, which do not have a `nocompatibility` level.

```
\MT@requires@pdftex6{
  \def\MT@define@optionX@#1#2{%
    \define@key{MTX}{#1}[true]{%
      \KV@@sp@def\@tempb{#1}%
      \MT@map@clist@n{##1}{%
        \KV@@sp@def\MT@val{####1}%
        \edef\@tempb{\csname MT@rbba@\@tempb\endcsname}%
        \MT@ifempty\MT@val\relax{%
          \@tempcnta=\m@ne
          \MT@ifstreq\MT@val{true}{%
            \MT@checksetup\@tempb{%
              \@tempcnta=\@ne
              \MT@info{Enabling #1}%
            }%
          }{%
            \MT@ifstreq\MT@val{false}{%
              \@tempcnta=\z@
              \MT@info{Disabling #1}%
            }{%
              \MT@warning{%
                Value `\MT@val' for key `#1' not recognised.\MessageBreak
                Use either `true' or `false'}%
            }%
          }%
          \ifnum\@tempcnta>\m@ne
            #2\relax
          \fi
        }%
      }%
    }%
  }
  \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@
                                \MT@trackingfalse\else\MT@trackingtrue\fi}
  \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
  \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
                               \pdfappendkern \@tempcnta}
}{
```

Disable for older pdfTEX versions.

```
  \define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
  \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
  \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
}
\define@key{MTX}{activate}[true]{%
  \setkeys{MTX}{protrusion={#1}}%
  \setkeys{MTX}{expansion={#1}}%
}
```

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

```
\let\MT@saved@setupfont\MT@setupfont
\define@key{MTX}{disable}[]{%
  \MT@info{Inactivate `\MT@MT' package}%
  \let\MT@setupfont\@gobble
}
\define@key{MTX}{enable}[]{%
  \MT@info{Reactivate `\MT@MT' package}%
  \let\MT@setupfont\MT@saved@setupfont
}
```

14.4.5 Processing the Options

\MT@ProcessOptionsWithKV Parse options.

```
⟨/package⟩
\def\MT@ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\MT@temp\@empty
  \MT@map@clist@c\@classoptionslist{%
    \def\CurrentOption{##1}%
    \MT@ifdefined@n@T{KV@#1@\CurrentOption}{%
      \edef\MT@temp{\MT@temp,\CurrentOption,}%
      \@expandtwoargs\@removeelement\CurrentOption
        \@unusedoptionlist\@unusedoptionlist
    }%
  }%
  \edef\MT@temp{%
    \noexpand\setkeys{#1}{%
      \MT@temp\@ptionlist{\@currname.\@currext}%
    }%
  }%
  \MT@temp
  \AtEndOfPackage{\let\@unprocessedoptions\relax}%
  \let\CurrentOption\@empty
}
\MT@ProcessOptionsWithKV{MT}
⟨*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 3230 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}
  \MT@protrusionfalse
  \MT@expansionfalse
  \MT@trackingfalse
  \MT@spacingfalse
  \MT@kerningfalse
  \MT@babelfalse
  \let\MT@setupfont\relax
  \def\DeclareMicrotypeSet#1#{\@gobbletwo}
  \renewcommand*\UseMicrotypeSet[2][]{}
  \renewcommand*\SetProtrusion[3][]{}
  \renewcommand*\SetExpansion[3][]{}
  \renewcommand*\SetTracking[3][]{}
  \renewcommand*\SetExtraSpacing[3][]{}
  \renewcommand*\SetExtraKerning[3][]{}
  \renewcommand*\DeclareCharacterInheritance[3][]{}
  \renewcommand*\DeclareMicrotypeAlias[2]{}
  \renewcommand*\LoadMicrotypeFile[1]{}
  \renewcommand*\microtypesetup[1]{}
  \renewcommand*\microtypecontext[1]{}
\else
```

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

```
  \ifnum\pdfoutput<\@ne
    \ifMT@opt@expansion \else
      \MT@expansionfalse
    \fi
  \fi
```

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

Some packages depend on the value of `\pdfoutput` and will get confused if it is changed after they have been loaded. These packages are, among others: `color`, `graphics`, `hyperref`, `crop`, `contour`, `pstricks` and, as a matter of course, `ifpdf`. Instead of testing for each package (that's not our job), we only say that it was `microtype` that changed it. This must be sufficient!

```
  \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
              \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

Fix the font sets.

```
  \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
```

Protrusion.

```
  \ifMT@protrusion
    \edef\MT@active@features{\MT@active@features,pr}
    \pdfprotrudechars\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
```

Expansion.

```
\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
    \ifnum\MT@stretch>\z@
      \let\MT@shrink\MT@stretch
    \else
      \let\MT@shrink\MT@shrink@default
    \fi
  \fi
```

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

```
  \ifnum\MT@step=\m@ne
    \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
  \else
    \@tempcnta=\MT@step
    \ifnum\@tempcnta=\z@
      \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
        Setting it to one}
    \fi
  \fi
  \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
  \edef\MT@step{\number\@tempcnta\space}
```

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

```
  \let\MT@auto\@empty
  \ifMT@auto
    \MT@requires@pdftex4{%
```

We turn off automatic expansion if output mode is DVI.

```
      \ifnum\pdfoutput<\@ne
        \ifMT@opt@auto
          \MT@warning@nl{%
```

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

Also, if pdfTEX is too old.

```
    }{%
      \MT@warning@nl{%
        The pdftex version you are using is too old for\MessageBreak
        automatic font expansion. I will switch it off and\MessageBreak
        hope that expanded fonts are available.\MessageBreak
        Otherwise, install pdftex version 1.20 or newer}
      \MT@autofalse
      \def\MT@auto{1000 }
    }
```

No automatic expansion.

```
  \else
    \ifnum\MT@pdftex@no < 4
      \def\MT@auto{1000 }
    \fi
  \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 an 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}%
  \pdfadjustspacing\MT@ex@level
  \MT@info@nl{\ifMT@auto\else Non-\fi Automatic 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@active@set{ex}
```

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

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

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

```
  \gdef\showhyphens#1{\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
}
\MT@requires@pdftex6{
⟨/package⟩
\MT@addto@setup{%
⟨*package⟩
```

Tracking, spacing and kerning.

```
    \ifMT@tracking
      \edef\MT@active@features{\MT@active@features,tr}
      \MT@info@nl{Tracking enabled}
      \MT@check@active@set{tr}
    \else
      \let\MT@tracking\relax
      \MT@info@nl{No tracking}
    \fi
    \ifMT@spacing
      \edef\MT@active@features{\MT@active@features,sp}
      \pdfadjustinterwordglue\@ne
      \MT@info@nl{Adjustment of interword spacing enabled}
      \MT@check@active@set{sp}
    \else
      \let\MT@spacing\relax
      \MT@info@nl{No adjustment of interword spacing}
    \fi
    \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
      \MT@info@nl{No adjustment of character kerning}
    \fi
⟨/package⟩
    \ifnum\MT@letterspace=\m@ne
      \let\MT@letterspace\MT@letterspace@default
    \else
      \MT@ls@too@large\MT@letterspace
    \fi
  }
```

If pdfTEX is too old, we disable tracking, spacing and kerning.

```
⟨*package⟩
}{
  \MT@addto@setup{%
    \ifMT@tracking
      \MT@warning@nl{Tracking only works with pdftex version 1.40\MessageBreak
        or newer. Switching it off}%
    \else
      \MT@info@nl{No tracking (pdftex too old)}
    \fi
    \MT@trackingfalse
    \let\MT@tracking\relax
    \ifMT@spacing
      \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
```

```
        pdftex version 1.40 or newer. Switching it off}%
    \else
      \MT@info@nl{No adjustment of interword spacing (pdftex too old)}
    \fi
    \MT@spacingfalse
    \let\MT@spacing\relax
    \ifMT@kerning
      \MT@warning@nl{Character kerning only works with\MessageBreak
        pdftex version 1.40 or newer. Switching it off}%
    \else
      \MT@info@nl{No adjustment of character kerning (pdftex too old)}
    \fi
    \MT@kerningfalse
    \let\MT@kerning\relax
  }
}
```

Warning if `\nonfrenchspacing` is active, since space factors will be ignored with `\pdfadjustinterwordglue` > 0. Why 1500? Because some packages redefine `\frenchspacing`. See the `c.t.t` thread '`\frenchspacing` with AMS packages and babel', started by Philipp Lehman on August 16, 2005: <`ddtbaj$rob$1@online.de`>.

```
\MT@requires@pdftex6{
  \AtBeginDocument{%
    \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
  }
}\relax
```

`\DisableLigatures` is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```
\MT@requires@pdftex5{
  \MT@addto@setup{%
    \ifMT@noligatures \else
      \let\MT@noligatures\relax
    \fi
  }
}\relax
```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```
\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}%
    \shorthandoff{#2}}
}{
  \def\MT@shorthandoff#1#2{%
    \MT@error{You must load the `babel' package before microtype}
             {Otherwise, I cannot switch off the active characters
              for #1 babel.}}
}
```

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}%
      }
```

Disable French `babel`'s active characters.

```
      \ifMT@kerning
        \MT@if@false
        \MT@with@babel@and{french}  \MT@if@true
        \MT@with@babel@and{frenchb} \MT@if@true
        \MT@with@babel@and{francais}\MT@if@true
        \MT@with@babel@and{acadian} \MT@if@true
        \MT@with@babel@and{canadien}\MT@if@true
        \ifMT@if@\MT@shorthandoff{French}{;:!?}\fi
```

Disable Turkish `babel`'s active characters.

```
        \MT@with@babel@and{turkish}{\MT@shorthandoff{Turkish}{:!=}}
      \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.

```
\AtBeginDocument\selectfont
```

\MT@curr@file This is the current file (hopefully with the correct extension).

```
\edef\MT@curr@file{\jobname.tex}
```

That was that.

```
⟨/package⟩
⟨/package|letterspace⟩
```

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,LY1,OT4,QX,T5,TS1,OML,OMS,U} }

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

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

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

\DeclareMicrotypeSet{smallcaps}
   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
     shape    = {sc}
   }

\DeclareMicrotypeSet{footnotesize}
   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
     size     = {-small}
   }

\DeclareMicrotypeSet{scriptsize}
```

```
  { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
    size     = {-footnotesize}
  }

\DeclareMicrotypeSet{normalfont}
  { font = */*/*/*/* }
```

The default sets.

```
%%% ----------------------------------------------------------------------
%%% DEFAULT SETS

\DeclareMicrotypeSetDefault[protrusion]{alltext}
\DeclareMicrotypeSetDefault[expansion] {basictext}
\DeclareMicrotypeSetDefault[spacing]   {basictext}
\DeclareMicrotypeSetDefault[kerning]   {alltext}
\DeclareMicrotypeSetDefault[tracking]  {normalfont}
```

15.2 Font Aliases

Fonts that are ‘the same’.

```
%%% ----------------------------------------------------------------------
%%% FONT ALIASES
```

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.

```
\DeclareMicrotypeAlias{lmr} {cmr}  % lmodern
\DeclareMicrotypeAlias{aer} {cmr}  % ae
\DeclareMicrotypeAlias{zer} {cmr}  % zefonts
\DeclareMicrotypeAlias{cmor}{cmr}  % eco
\DeclareMicrotypeAlias{hfor}{cmr}  % hfoldsty
```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the qfonts package, which provides both font families.

```
\DeclareMicrotypeAlias{pxr} {ppl}  % pxfonts
\DeclareMicrotypeAlias{qpl} {ppl}  % qfonts/QuasiPalatino
\DeclareMicrotypeAlias{txr} {ptm}  % txfonts
\DeclareMicrotypeAlias{qtm} {ptm}  % qfonts/QuasiTimes
```

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

The eulervm package virtually extends the Euler fonts.

```
\DeclareMicrotypeAlias{zeur}{eur}  % Euler VM
\DeclareMicrotypeAlias{zeus}{eus}
```

The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

```
\DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
\DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond
```

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
   {french,francais,acadian,canadien}
   {kerning=french, spacing=}

\DeclareMicrotypeBabelHook
   {english,UKenglish,british,american,USenglish}
   {kerning=, spacing=nonfrench}

\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 \mathchardefed 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.

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},
     - = {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},
  }

```

15.5.5 QX

The Central European QX encoding. 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},
     S = {\'S,\c S,\v S},
     s = {\'s,\c s,\v s},
     T = {\c T},
     t = {\c 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},
   }

 ⟨/m-t⟩
```

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```
⟨*zpeu⟩
\DeclareCharacterInheritance
  { encoding = U,
    family   = {zpeu,zpeus,eurosans} }
  { E = 128 }

⟨/zpeu⟩
⟨*mvs⟩
\DeclareCharacterInheritance
  { encoding = OT1,
    family   = mvs }
  { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm

```

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 = U,
    family   = mvs }
  { 164 = {099,100,101} }

⟨/mvs⟩
```

15.6 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).

```
⟨*m-t⟩
%%% -----------------------------------------------------------------------
%%% EXPANSION SETTINGS

\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,
  }

```

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⟩
```

15.7 Character Protrusion

```
%%% -----------------------------------------------------------------------
%%% PROTRUSION SETTINGS

```

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},
   }
```

15.7.1 Default

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 ]
```

Computer Modern Roman (cmr),

```
⟨cmr⟩    [ name     = cmr-default ]
```

Adobe Garamond (pad, padx, padj),

```
⟨pad⟩    [ name     = pad-default ]
```

Minion[13] (pmnx, pmnj),

```
⟨pmn⟩    [ name     = pmnj-default ]
```

Palatino (ppl, pplx, pplj),

```
⟨ppl⟩    [ name     = ppl-default ]
```

Times (ptm, ptmx, ptmj),

```
⟨ptm⟩    [ name     = ptm-default ]
```

and URW Garamond (ugm).

```
⟨ugm⟩    [ name     = ugm-default ]
```

[13] Contributed by Harald Harders (h.harders@tu-bs.de)

```
⟨m-t⟩   { encoding = OT1     }
⟨cmr⟩   { }
⟨bch|pad|pmn|ugm⟩   { encoding = OT1,
⟨ppl|ptm⟩   { encoding = {OT1,OT4},
⟨bch⟩     family   = bch }
⟨pad⟩     family   = {pad,padx,padj} }
⟨pmn⟩     family   = pmnj }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
⟨ugm⟩     family   = ugm }
  {
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     A = {50,50},
⟨ugm⟩     A = {50,100},
⟨m-t|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},
⟨m-t|cmr|pad|pmn|ppl|ptm|ugm⟩     J = {50,  },
⟨bch⟩     J = {100,  },
    K = {  ,50},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     L = {  ,50},
⟨ptm⟩     L = {  ,80},
⟨ugm⟩     L = {  ,120},
⟨bch|pad|pmn|ugm⟩     O = {50,50},
⟨pad|pmn⟩   \OE = {50,  },
⟨ugm⟩   \OE = {50,50},
⟨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},
⟨ugm⟩     T = {70,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     V = {50,50},
⟨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},
⟨ptm|ugm⟩     Y = {80,80},
⟨ugm⟩     Z = {50,50},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     k = {  ,50},
⟨ugm⟩     k = {  ,70},
⟨pmn⟩     l = {  ,-50},
⟨pad|ppl⟩     p = {50,50},
⟨ugm⟩     p = {  ,50},
⟨pad|ppl⟩     q = {50,  },
    r = {  ,50},
⟨cmr|pad|pmn⟩     t = {  ,70},
⟨bch⟩     t = {  ,50},
⟨ugm⟩     t = {  ,100},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     v = {50,50},
⟨ugm⟩     v = {50,70},
⟨m-t|bch|cmr|pad|pmn|ppl|ptm⟩     w = {50,50},
⟨ugm⟩     w = {50,70},
     x = {50,50},
⟨m-t|bch|pad|pmn⟩     y = {  ,50},
⟨cmr|ppl|ptm⟩     y = {50,70},
⟨ugm⟩     y = {  ,70},

⟨cmr⟩     0 = {  ,50},
⟨m-t⟩     1 = {50,50},
⟨bch|pad|ptm|ugm⟩     1 = {150,150},
⟨cmr⟩     1 = {100,200},
⟨pmn⟩     1 = {  ,50},
⟨ppl⟩     1 = {100,100},
⟨bch|cmr|pad|ugm⟩     2 = {50,50},
⟨cmr|pad|ugm⟩     3 = {50,50},
⟨bch|pmn⟩     3 = {50,  },
⟨m-t|pad⟩     4 = {50,50},
⟨bch⟩     4 = {100,50},
⟨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},
⟨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},
    {,}= { ,500},
⟨m-t|cmr|pad|pmn|ppl|ptm|ugm⟩     : = { ,500},
⟨bch⟩     : = { ,400},
⟨m-t|bch|pad|pmn|ptm⟩     ; = { ,300},
⟨cmr|ppl⟩     ; = { ,500},
⟨ugm⟩     ; = { ,400},
     ! = { ,100},
⟨m-t|pad|pmn|ptm⟩     ? = { ,100},
⟨bch|cmr|ppl|ugm⟩     ? = { ,200},
⟨pmn⟩     " = {300,300},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     @ = {50,50},
⟨ptm⟩     @ = {100,100},
⟨m-t|bch|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},
⟨m-t|ppl|ptm|ugm⟩     * = {200,200},
⟨bch|pmn⟩     * = {200,300},
⟨cmr|pad⟩     * = {300,300},
⟨m-t|cmr|ppl|ptm⟩     + = {250,250},
⟨bch⟩     + = {150,250},
⟨pad⟩     + = {300,300},
⟨pmn⟩     + = {150,200},
⟨ugm⟩     + = {250,300},
⟨ugm⟩    {=}= {200,200},
⟨m-t|pad|pmn|ptm⟩     ( = {100,   },    ) = {   ,200},
⟨bch|ugm⟩     ( = {200,   },    ) = {   ,200},
⟨cmr|ppl⟩     ( = {100,   },    ) = {   ,300},
⟨bch|pmn⟩     [ = {100,   },    ] = {   ,100},

⟨m-t|pad|pmn|ptm⟩     / = {100,200},
⟨bch⟩     / = { ,200},
⟨cmr|ppl⟩     / = {200,300},
⟨ugm⟩     / = {100,300},
⟨m-t|ptm⟩     - = {500,500},
⟨bch|cmr|ppl⟩     - = {400,500},
⟨pad⟩     - = {300,500},
⟨pmn⟩     - = {200,400},
⟨ugm⟩     - = {500,600},
⟨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},
⟨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},
⟨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.

```
⟨*cmr⟩
\SetProtrusion
   [ name     = cmr-OT1,
     load     = cmr-default ]
   { encoding = {OT1,OT4},
     family   = cmr  }
   {
     \AE = { 50,   },
     "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⟩
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```
\SetProtrusion
⟨m-t⟩   [ name     = T1-default,
⟨bch⟩   [ name     = bch-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 ]
⟨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}    }
⟨bch|cmr|pad|pmn|ppl⟩   { encoding = {T1,LY1},
⟨ptm|ugm⟩   { encoding = {T1},
⟨bch⟩     family   = bch }
⟨cmr⟩     family   = cmr }
⟨pad⟩     family   = {pad,padx,padj} }
⟨pmn⟩     family   = pmnj }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
⟨ugm⟩     family   = ugm }
   {
⟨cmr⟩     \AE = {50,  },
⟨bch⟩     \OE = {50,  },
⟨pmn⟩     \TH = {  ,50},
⟨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},
⟨cmr|ppl⟩     \textbackslash    = {200,300},
⟨ugm⟩     \textbackslash    = {100,300},
⟨ugm⟩     \textbar          = {200,200},
⟨cmr⟩     \textquotedblleft = {200,600},
⟨cmr⟩     \textquotedbl     = {300,300},
```

The EC fonts do something weird: they insert an implicit kern between quote and boundary character. Therefore, we must override the settings from OT1.

```
⟨m-t|cmr|pad|ppl|ptm|ugm⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨bch|pmn⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨m-t|bch|pmn⟩     \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨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},
⟨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,   },
⟨ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨m-t|cmr|pad|ppl|ptm|ugm⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨bch|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, on the other hand, restore the original kerning from the OT1 fonts, and so do we. Silly, isn't it?

```
⟨*cmr⟩
\SetProtrusion
   [ name     = lmr-T1,
     load     = cmr-T1   ]
   { encoding = {T1,LY1},
     family   = lmr      }
   {
     \textquotedblleft = {500,300},
     \quotedblbase     = {500,300},
   }

⟨/cmr⟩
```

Settings for the QX encoding (generic and Times). 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} }
   {
⟨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 }
        {
⟨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⟩
⟨*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},
    }

⟨/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⟩
```

15.7.2 Italics

To find default settings for italic is difficult, since the character shapes and their behaviour at the beginning or end of line may be wildly different for different fonts. Therefore, we leave the letters away, and only set up the punctuation characters.

```
\SetProtrusion
⟨m-t⟩    [ name     = OT1-it   ]
⟨bch⟩    [ name     = bch-it   ]
⟨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|pad|pmn|ugm⟩   { encoding = OT1,
⟨ppl|ptm⟩   { encoding = {OT1,OT4},
⟨bch⟩      family   = bch,
⟨pad⟩      family   = {pad,padx,padj},
⟨pmn⟩      family   = pmnj,
⟨ppl⟩      family   = {ppl,pplx,pplj},
⟨ptm⟩      family   = {ptm,ptmx,ptmj},
⟨ugm⟩      family   = ugm,
⟨!(cmr|ugm)⟩     shape    = {it,sl}  }
⟨ugm⟩      shape    = it  }
⟨cmr⟩    { }
  {
⟨cmr|ptm⟩     A = {100,50},
⟨pad|pmn⟩     A = {50,  },
⟨ugm⟩     A = {  ,150},
⟨ppl⟩     A = {50,50},
⟨ptm⟩   \AE = {100,  },
⟨pad|ppl⟩   \AE = {50,  },
⟨pmn⟩   \AE = {  ,-50},
⟨cmr|pad|ppl|ptm⟩     B = {50,  },
⟨pmn⟩     B = {20,-50},
⟨bch|ppl|ptm|ugm⟩     C = {50,  },
⟨cmr|pad⟩     C = {100, },
⟨pmn⟩     C = {50,-50},
⟨cmr|pad|ppl|ptm⟩     D = {50,50},
⟨pmn⟩     D = {20,  },
⟨cmr|pad|ppl|ptm⟩     E = {50,  },
⟨pmn⟩     E = {20,-50},
⟨cmr|pad|ptm⟩     F = {100, },
⟨pmn⟩     F = {10,  },
⟨ppl⟩     F = {50,  },
⟨bch|ppl|ptm|ugm⟩     G = {50,  },
⟨cmr|pad⟩     G = {100, },
⟨pmn⟩     G = {50,-50},
⟨cmr|pad|ppl|ptm⟩     H = {50,  },
⟨cmr|pad|ptm⟩     I = {50,  },
⟨pmn⟩     I = {20,-50},
⟨cmr|ptm⟩     J = {100, },
⟨pad⟩     J = {50,  },
⟨pmn⟩     J = {20,  },
⟨cmr|pad|ppl|ptm⟩     K = {50,  },
⟨pmn⟩     K = {20,  },
⟨cmr|pad|ppl|ptm⟩     L = {50,  },
⟨pmn⟩     L = {20,50},
⟨ugm⟩     L = {  ,100},
⟨cmr|ptm⟩     M = {50,  },
⟨pmn⟩     M = {  ,-30},
⟨cmr|ptm⟩     N = {50,  },
⟨pmn⟩     N = {  ,-30},
⟨bch|pmn|ppl|ptm⟩     O = {50,  },
⟨cmr|pad⟩     O = {100, },
⟨ugm⟩     O = {70,50},
⟨pmn|ppl|ptm⟩    \OE = {50,  },
⟨pad⟩    \OE = {100, },
⟨cmr|pad|ppl|ptm⟩     P = {50,  },
⟨pmn⟩     P = {20,-50},
⟨bch|pmn|ppl|ptm⟩     Q = {50,  },
⟨cmr|pad⟩     Q = {100, },
⟨ugm⟩     Q = {70,50},
⟨cmr|pad|ppl|ptm⟩     R = {50,  },
⟨pmn⟩     R = {20,  },
⟨bch|cmr|pad|ppl|ptm⟩     S = {50,  },
⟨pmn⟩     S = {20,-30},
⟨bch|cmr|pad|ppl|ptm⟩     $ = {50,  },
⟨pmn⟩     $ = {20,-30},
⟨bch|pmn|ugm⟩     T = {70,  },
⟨cmr|pad|ppl|ptm⟩     T = {100, },
⟨cmr|pad|ppl|ptm⟩     U = {50,  },
⟨pmn⟩     U = {50,-50},
⟨cmr|pad|pmn|ugm⟩     V = {100, },
⟨ppl|ptm⟩     V = {100,50},
⟨cmr|pad|pmn|ugm⟩     W = {100, },
⟨ppl⟩     W = {50,  },
⟨ptm⟩     W = {100,50},
⟨cmr|ppl|ptm⟩     X = {50,  },
⟨cmr|ptm⟩     Y = {100, },
⟨pmn⟩     Y = {50,  },
⟨ppl⟩     Y = {100,50},
⟨pmn⟩     Z = {  ,-50},
⟨pmn⟩     d = {  ,-50},
⟨pad|pmn⟩     f = { ,-100},
⟨pmn⟩     i = {  ,-30},
⟨pmn⟩     j = {  ,-30},
⟨pmn⟩     l = {  ,-100},
⟨bch⟩     o = {50,50},
⟨bch⟩     p = {  ,50},
⟨pmn⟩     p = {-50,  },
⟨bch⟩     q = {50,  },
⟨pmn⟩     r = {  ,50},
⟨bch⟩      t = {  ,50},
⟨pmn|ugm⟩      v = {50,  },
⟨bch⟩      w = {  ,50},
⟨pmn|ugm⟩      w = {50,  },
⟨bch⟩      y = {  ,50},
⟨cmr⟩      0 = {100, },
⟨bch|ptm⟩      1 = {150,100},
⟨cmr⟩      1 = {200,50},
⟨pad⟩      1 = {150, },
⟨pmn⟩      1 = {50,  },
⟨ppl⟩      1 = {100, },
⟨ugm⟩      1 = {150,150},
⟨cmr⟩      2 = {100,-100},
⟨pad|ppl|ptm⟩      2 = {50,  },
⟨pmn⟩      2 = {-50,  },
⟨bch⟩      3 = {50,  },
⟨cmr⟩      3 = {100,-100},
⟨pmn⟩      3 = {-100, },
⟨ptm⟩      3 = {100,50},
⟨bch⟩      4 = {100, },
⟨cmr|pad⟩      4 = {150, },
⟨ppl|ptm⟩      4 = {50,  },
⟨cmr⟩      5 = {100, },
⟨ptm⟩      5 = {50,  },
⟨bch⟩      6 = {50,  },
⟨cmr⟩      6 = {100, },
⟨bch|pad|ptm⟩      7 = {100, },
⟨cmr⟩      7 = {200,-150},
⟨pmn⟩      7 = {20,  },
⟨ppl⟩      7 = {50,  },
⟨cmr⟩      8 = {50,-50},
⟨cmr⟩      9 = {100,-100},
⟨m-t|cmr|pad|pmn|ppl⟩      . = { ,500},
⟨bch|ptm|ugm⟩      . = { ,700},
⟨m-t|cmr|pad|pmn|ppl⟩     {,}= { ,500},
⟨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⟩      & = {100,50},
⟨ugm⟩      & = {50,100},
⟨m-t|cmr|pad|pmn⟩     \% = {100, },
⟨bch⟩     \% = {50,50},
⟨ppl|ptm⟩     \% = {100,100},
⟨ugm⟩     \% = {100,50},
⟨m-t|pmn|ppl⟩      * = {200,200},
⟨bch⟩      * = {300,200},
⟨cmr⟩      * = {400,100},
⟨pad⟩      * = {500,100},
⟨ptm|ugm⟩      * = {400,200},
⟨m-t|cmr|pmn|ppl⟩     + = {150,200},
⟨bch|ugm⟩     + = {250,250},
⟨pad|ptm⟩     + = {250,200},
⟨m-t|pad|pmn|ppl⟩     @ = {50,50},
⟨bch⟩     @ = {80,50},
⟨cmr⟩     @ = {200,50},
⟨ptm⟩     @ = {150,150},
⟨m-t|bch|ugm⟩     ~ = {150,150},
⟨cmr|pad|pmn|ppl|ptm⟩     ~ = {200,150},
⟨ugm⟩    {=}= {200,200},
      ( = {200, },    ) = {  ,200},
⟨m-t|cmr|pad|ppl|ptm|ugm⟩     / = {100,200},
⟨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},
⟨m-t|pmn⟩     \textendash       = {200,200},   \textemdash        = {150,150},
⟨bch⟩     \textendash       = {200,300},   \textemdash        = {150,200},
⟨cmr⟩     \textendash       = {500,300},   \textemdash        = {400,200},
⟨pad|ppl|ptm|ugm⟩     \textendash       = {300,300},   \textemdash        = {200,200},
⟨m-t|bch|pmn|ugm⟩     \textquoteleft    = {400,200},   \textquoteright    = {400,200},
⟨cmr|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},
⟨cmr⟩     \textquotedblleft = {700,100},   \textquotedblright = {500,300},
⟨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⟩
\SetProtrusion
   [ name     = cmr-it-OT1,
     load     = cmr-it   ]
   { encoding = {OT1,OT4},
     family   = cmr,
     shape    = it       }
   {
     \AE = {100,   },
     \OE = {100,   },
     "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
     "0A = { 50, 50}, % \Omega
   }

⟨/cmr⟩
\SetProtrusion
⟨m-t⟩   [ name     = T1-it-default,
⟨bch⟩  [ name     = bch-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   ]
⟨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},
⟨ptm|ugm⟩  { encoding = {T1},
⟨bch⟩    family   = bch,
⟨cmr⟩    family   = cmr,
⟨pmn⟩    family   = pmnj,
⟨pad⟩    family   = {pad,padx,padj},
⟨ppl⟩    family   = {ppl,pplx,pplj},
⟨ptm⟩    family   = {ptm,ptmx,ptmj},
⟨ugm⟩    family   = ugm,
⟨!(cmr|ugm)⟩    shape    = {it,sl}  }
⟨cmr|ugm⟩    shape    = it       }
   {
⟨m-t|bch|pmn⟩    _ = {  ,100},
⟨cmr|ugm⟩    _ = {100,200},
⟨pad|ppl|ptm⟩    _ = {100,100},
⟨cmr⟩    \AE = {100,   },
⟨bch⟩    \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},
⟨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|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-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 = {700,100},
     \quotedblbase     = {600,300},
   }

```

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},
  }

⟨/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⟩
```

15.7.3 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).

```
⟨*!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},
   }

⟨/!ugm⟩
⟨*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},
  }

```

15.7.4 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⟩
```

15.7.5 textcomp

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```
\SetProtrusion
⟨m-t⟩   [ name     = textcomp ]
⟨bch⟩   [ name     = bch-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 }
⟨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 }
   {
⟨cmr⟩     \textquotestraightbase    = {300,300},
⟨pad|pmn⟩     \textquotestraightbase    = {400,400},
⟨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},
⟨cmr|pmn⟩     \textquotesingle          = {300,400},
⟨pad⟩     \textquotesingle          = {400,500},
⟨ptm⟩     \textquotesingle          = {500,500},
⟨ugm⟩     \textquotesingle          = {300,500},
⟨bch|cmr|pmn⟩     \textasteriskcentered     = {200,300},
⟨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},
⟨ugm⟩     \textminus                = {250,300},
⟨bch|pad|pmn⟩     \textlbrackdbl            = {100,   },
⟨bch|pad|pmn⟩     \textrbrackdbl            = {   ,100},
⟨pmn⟩     \textasciigrave           = {200,500},
⟨bch|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},
⟨cmr|pad⟩     \textdagger               = {100,100},
⟨ptm⟩     \textdagger               = {150,150},
⟨cmr|pad|pmn⟩     \textdaggerdbl            = { 80, 80},
⟨ptm⟩     \textdaggerdbl            = {100,100},
⟨bch⟩     \textbardbl               = {100,100},
⟨ugm⟩     \textbardbl               = {150,150},
⟨bch⟩     \textbullet               = {200,200},
⟨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},
⟨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},
⟨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},
⟨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,   },
⟨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},
⟨cmr|pad⟩     \textdegree               = {400,400},
⟨pmn⟩     \textdegree               = {150,400},
⟨ugm⟩     \textdegree               = {200,200},
⟨bch|cmr|pad|pmn|ugm⟩     \textpm                   = {150,200},
⟨ptm⟩     \textpm                   = { 50, 80},
⟨bch|ugm⟩     \texttwosuperior          = {100,200},
⟨cmr⟩     \texttwosuperior          = { 50,100},
⟨pad|pmn⟩     \texttwosuperior          = {200,200},
⟨ptm⟩     \texttwosuperior          = { 50, 50},
⟨bch|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},
⟨ptm⟩     \textperiodcentered       = {300,300},
⟨ugm⟩     \textperiodcentered       = {200,500},
⟨bch|ugm⟩     \textonesuperior          = {200,300},
⟨cmr|pad|pmn⟩     \textonesuperior          = {200,200},
⟨ptm⟩     \textonesuperior          = {100,100},
⟨bch|pad|pmn|ugm⟩     \textordmasculine         = {200,200},
⟨cmr⟩     \textordmasculine         = {100,200},
⟨bch|cmr|pmn⟩     \texteuro                 = {100,   },
⟨pad⟩     \texteuro                 = { 50,100},
⟨bch|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},
⟨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⟩
```

15.7.6 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⟩
```

15.7.7 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⟩
```

15.7.8 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⟩
```

15.7.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym[14]).

```
⟨*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
   { encoding = U,
     family   = {zpeus,eurosans} }
   {
     E = {100,50}
   }

\SetProtrusion
   { encoding = U,
     family   = {zpeus,eurosans},
     shape    = it* }
   {
     E = {200, }
   }

⟨/zpeu⟩
⟨/cfg-e⟩
```

15.8 Interword Spacing

Default unit is space.

```
⟨*m-t⟩
%%% -----------------------------------------------------------------------
%%% INTERWORD SPACING SETTINGS

\SetExtraSpacing
   [ name = default ]
```

[14] Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

```
{ encoding = {OT1,T1,LY1,OT4,QX,T5} }
{
```

These settings are only a first approximation. The following reasoning is from a mail from *Ulrich Dirr*. I do not claim to have coped with the task. (. . . In fact, I think these settings are wrong. They lead to more overfull boxes than without spacing adjustment. Needs to be fixed.)

> ‘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 of 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 of 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.’

```
   }

```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a `factor` < 1000.)
- Should the first parameter also be used? (Probably.)

15.8.1 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
```

```
     . = {333,2000,-667},
```

```
  \sfcode`\? 3000
```

```
     ? = {333,2000,-667},
```

```
  \sfcode`\! 3000
```

```
     ! = {333,2000,-667},
```

```
  \sfcode`\: 2000
```

```
     : = {333,1000,-500},
```

```
  \sfcode`\; 1500
```

```
    ; = {   , 500,-333},
```

```
\sfcode`\, 1250
```

```
   {,}= {   , 250,-200},
```

```
}
```

```
  }

```

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.

```
\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},
  }

```

15.9 Additional Kerning

Default unit is 1em.

```
%%% ---------------------------------------------------------------------
%%% ADDITIONAL KERNING

```

A dummy list to be loaded when no context is active.

```
\SetExtraKerning
  [ name = empty ]
  { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
  { }

```

15.9.1 French

For the French context.

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

```

This has the disadvantage that the 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     }
   { }

```

15.9.2 Turkish

And for Turkish.

```
\SetExtraKerning
   [ name     = turkish,
     context  = turkish,
     unit     = space  ]
   { encoding = {OT1,T1,LY1} }
   {
     :  = {500, }, % ~ \thinspace
     !  = {500, },
    {=} = {500, },
   }

⟨/m-t⟩
⟨/config⟩
```

16 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 Change History

Version 1.0 (2004/09/11)

General: Initial version 1

Version 1.1 (2004/09/21)

General: configuration file names in lowercase (suggested by *Harald Harders*) 65
issue an error instead of a warning, when pdfTEX version is too old for `autoexpand` 107
remove 8-bit characters from the configuration files (suggested by *Harald Harders*) 113
Protrusion: add factors for some more characters 118
settings for Adobe Minion (contributed by *Harald Harders*) 119
`\DeclareCharacterInheritance`: new command: possibility to specify character inheritance . . 91
`\MT@declare@sets`: remove spaces around set name 80
`\MT@DeclareSet@`: remove spaces around first argument 79
`\MT@find@file`: fix: also check whether the file for the base font family has already been loaded 65
`\MT@get@basefamily`: only remove suffix, if it is 'x' or 'j' 67
`\MT@get@listname@`: don't check for empty attributes list 67
`\MT@ifempty`: fix: use category code 12 for the percent character (reported by *Tom Kink*) 37
`\MT@is@number`: numbers may also be specified in hexadecimal or octal (suggested by *Harald Harders*) 71
`\MT@pdftex@no`: fix concerning version check (reported by *Harald Harders*) 32
`\MT@permute`: don't use sets for empty encoding . . . 93
`\MT@pr@split`: fix: allow zero and negative values . 50
`\MT@use@set`: remove spaces around set name 84
`\UseMicrotypeSet`: remove spaces around first argument 83

Version 1.2 (2004/10/03)

General: check for packages that might load fonts . 75
check whether only one encoding specified . . . 92
Font Sets: declare `cmor` as an alias of `cmr` 112
new: `allmath` and `basicmath` 111
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in `TS1` encoding 138
add settings for Computer Modern Roman math symbols 143
`\MT@context`: fix: set inheritance list `\globally` to `\@empty` 69
`\MT@familyalias`: define alias font name as an alternative, not as a replacement 47
`\MT@get@basefamily`: also remove 'w' (swash capitals) 67
`\MT@get@highlevel`: check whether defaults have changed 81
`\MT@get@listname@`: alternatively check for alias font name 67
`\MT@get@size`: additional magic to catch some errors 82
hijack `\set@fontsize` instead of `\@setfontsize` 82
`\MT@get@slot`: fix: group must also include `\MT@get@composite` 69
`\MT@loop`: fix: new macro, used instead of `\loop` . . 40
`\MT@maybe@do`: also check for alias font name 47
`\MT@permute@@@@@@`: more sanity checks for `\SetProtrusion` and `\SetExpansion` 94
`\MT@setupfont`: also search for alias font file 46
fix: call `\@@enc@update` if necessary 46

Version 1.3 (2004/10/27)

Font Sets: declare `aer`, `zer` and `hfor` as an alias of `cmr` 112
`\MT@catcodes`: check some category codes (compatibility with `german`) 33
`\MT@define@opt@keys`: fix: specifying `load` option does no longer require to give a `name`, too . . 89
`\MT@load@list`: check whether list exists 65

Version 1.4 (2004/11/12)

General: don't use scratch registers in global definitions 69
no need to check for packages that might load fonts anymore 75
use `\pickup@font` instead of `\define@newfont` as the hook for `\MT@setupfont` 75
use one instead of five counters 41
Protrusion: tweak quote characters for `cmr` variants (`OT1`, `T1`, `lmr`) 123
`\microtypesetup`: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options 102
`\MT@pdfcprot@error`: check for `pdfcprot` 43
`\SetExpansion`: fix: specifying extra options does no longer require to give a `name`, too 86

Version 1.4a (2004/11/17)

General: new option: final 99

\MT@normal@catcodes: fix: reset some more catcodes when reading files (reported by *Michael Hoppe*) 66

Version 1.4b (2004/11/26)

General: fix: set catcodes before reading global configuration file (reported by *Christoph Bier*) 101
- new message if \pdfoutput is changed 105
- optimisation: use less \expandafters and \csnames 36

Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl) 118
- slanted like italics 126

\MT@checklist@family: fix: don't try alias family name if encoding failed 48

\MT@get@basefamily: fix: failed for font names of the form abczz (reported by *Georg Verweyen*) 67

\MT@get@slot: don't define \MT@char globally (save stack problem) 69

\MT@ifdimen: don't set \MT@count globally (save stack problem) 38

\MT@use@set: don't use undeclared font sets 84

Version 1.5 (2004/12/15)

General: defaults: step: 4 (suggested by *Hàn Thế Thành*) 99
- defaults: calculate step as min(stretch,shrink)/5 106
- defaults: turn off expansion for DVI output 105
- disable automatic expansion for DVI output 106
- new option: selected, by default false (suggested by *Hàn Thế Thành*) 98

Documentation: add note about DVIoutput option 8
- add short history (section 13) 26

Inheritance: remove \ss from T1 list, add \DJ 114

Protrusion: settings for Bitstream Charter 119

\DeclareMicrotypeAlias: remove spaces around arguments 85

\MT@catcodes: reset catcode of '^' (compatibility with chemsym) 33

\MT@get@highlevel: don't test defaults if called after begin document 81

\MT@normal@catcodes: reset catcode of '=' (compatibility with Turkish babel) 66

\MT@scale@factor: warning for factors outside limits 52

\MT@scale@to@em: don't use \lpcode and \rpcode for the calculation 51

\MT@set@ex@codes: allow non-selected font expansion 56

\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters 49

Version 1.6 (2005/01/24)

General: defaults: turn off expansion for old pdfTEX versions 100
- disable automatic expansion for old pdfTEX versions 107
- load a font, if none is active 46
- new option: factor, by default 1000 99
- restructure dtx file 111
- test whether \pickup@font has changed 76
- use e-TEX's \ifcsname and \ifdefined if defined 36

Protrusion: add italic uppercase Greek letters 126
- improve settings for numbers (pointed out by *Peter Muthesius*) 121
- tune CMR math letters (OML encoding) 143

\MT@def@num@opt: test whether numeric options receive a number 99

\MT@get@charwd: use e-TEX's \fontcharwd, if available 51

\MT@get@inh@list: correct message if selected is false 69

\MT@set@ex@codes: introduce factor option 56

\MT@set@pr@codes: introduce factor option 49

\MT@use@set: retain current set if new set is undeclared 84

\MT@vinfo: new macro: used instead of \ifMT@verbose 30

Version 1.6a (2005/02/02)

Documentation: add table of fonts with tailored protrusion settings 18

\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem reported by *Bernard Gaulle*) 69

\MT@pdftex@no: new macro 32

\MT@reset@ef@codes: only reset \efcodes for older pdfTEX versions 56

Version 1.7 (2005/03/23)

- General: \SetExpansion: bug fix: remove space after autoexpand 91
  - \SetExpansion: don't allow automatic expansion for old pdfTeX versions 91
  - allow specification of size ranges (suggested by *Andreas Bühmann*) 81
  - modify \showhyphens 107
  - new value for verbose option: errors 99
  - shorter command names 41
  - warning when running in draft mode 105
- Documentation: add hint about compatibility 23
  - remove table of match order 12
- Protrusion: fix: remove \ from OT1, add \textbackslash to T1 encoding 122
- \DeclareMicrotypeAlias: may also be used inside configuration files 85
- \LoadMicrotypeFile: new command (suggested by *Andreas Bühmann*) 85
- \Microtype@Hook: new command for font package authors 101
- \microtypesetup: fix: warning also when setting to (no)compatibility 102
- \MT@begin@catcodes: also use inside configuration commands 66
- \MT@get@listname@: use \@tfor (*Andreas Bühmann*'s idea) 67
- \MT@get@slot: remove backslash hack 69
  - test for \chardefed commands 70
  - test whether \⟨*encoding*⟩\⟨...⟩ is defined 70
- \MT@if@list@exists: don't define \MT@#1@c@name \globally, here and elsewhere 68
- \MT@ifdimen: comparison with 1 to allow size smaller than 1 (suggested by *Andreas Bühmann*) 38
- \MT@increment: use e-TeX's \numexpr if available 41
- \MT@is@composite: new macro: construct command for composite character; no uncontrolled expansion 73
- \MT@normal@catcodes: reset catcode of ':' (compatibility with french* packages) 66
- \MT@scale: new macro: use e-TeX's \numexpr if available 41
- \MT@set@ex@codes: two versions of this macro 56
- \MT@split@name: don't define \MT@encoding &c. \globally 47
- \MT@test@ast: make it simpler 81
- \MT@try@order: always check for size, too (suggested by *Andreas Bühmann*) 67
  - fix: also check for //⟨*series*⟩/⟨*shape*⟩// (reported by *Andreas Bühmann*) 67
- \MT@warn@code@too@large: new macro: type out maximum protrusion factor 52
- \MT@warn@err: new macro: for verbose=errors 30

Version 1.8 (2005/06/23)

- General: \SetProtrusion: new key: unit 90
  - if font substitution has occurred, set up the substitute font, not the selected one 75
  - new option: config to load a different main configuration file 100
  - new option: unit, by default character 100
- Documentation: add example for factor option 13
  - add example of how to get rid of a widow (suggested by *Adam Kucharczyk*) 14
  - add hint about error messages 23
- Font Sets: add U encoding to allmath 111
  - declare pxr and txr as aliases of ppl resp. ptm 112
- Inheritance: remove \DJ from T1 list (it's the same as \DH) 114
- Protrusion: add LY1 characters for Times 125
  - settings for AMS math fonts 147
  - verified settings for slanted Computer Modern Roman 132
- \DeclareMicrotypeAlias: warning when overriding an alias font 85
- \DeclareMicrotypeSetDefault: new command: set default font set 84
- \MT@check@rlist: made recursive 95
- \MT@curr@list@name: new macro: current list type and name 73
- \MT@declare@sets: warning when redefining a set 80
- \MT@define@set@key@: use comma lists instead of token lists 80
- \MT@find@file: no longer wrap names in commands 65
- \MT@get@charwd: warning for missing (resp. zero-width) characters 51
- \MT@get@font@dimen@six: new macro: test whether \fontdimen 6 is defined 50
- \MT@get@listname@: made recursive 67
- \MT@get@slot: fix: expand active characters 70
  - test whether \⟨*encoding*⟩\⟨...⟩ is defined made more robust 70
- \MT@get@unit: new macro: get unit for codes 54
- \MT@in@rlist: made recursive 40
- \MT@is@active: new macro: translate inputenc-defined characters 71
- \MT@is@letter: warning for non-ASCII characters 70
- \MT@led@kern: character protrusion with ledmac 43
- \MT@make@string: use \@onelevel@sanitize 41
- \MT@map@clist@n: new macro: used instead of \@for 39
- \MT@map@tlist@n: new macro: used instead of \@tfor 39
- \MT@normal@catcodes: reset catcodes of the remaining ASCII characters 66
- \MT@old@cmd: renamed commands from \..MicroType.. to \..Microtype.. 30
- \MT@orig@add@accent: fix: disable micro-typographic setup inside \add@accent (reported by *Stephan Hennig*) 76
- \MT@pdftex@no: case 5: pdfTeX 1.30 32
- \MT@permute@@@@@@: add ranges to the beginning of the lists 94
- \MT@pr@split: get character width once only 50
- \MT@scale: fix: remove spaces in non-e-TeX variant (reported by *Mark Rossi*) 41
- \MT@setupfont@hook: restore \% and \# when hyperref is loaded 44

restore csquotes's active characters 44
restore percent character if Spanish babel is loaded 44
\MT@use@set: fix: remove braces in first line 84
\MT@xadd: simplified 39

Version 1.9 (2005/10/28)

General: \DeclareMicrotypeSet: new key: font 82
\SetProtrusion and \SetExpansion: new key: font 88
\SetProtrusion: value 'relative' renamed to 'character' for key unit 90
allow context-specific font setup 75
disable expansion if both step and shrink are zero 107
disable microtype setup inside hyperref's \pdfstringdef (reported by *Hàn Thế Thành*) 44
option unit: rename value relative to character 100
warning if user requested zero step 106
Documentation: add hint about verbatim environment 23
add remark about Type 1 fonts required for automatic font expansion 7
Font Sets: add OT4 encoding to text sets 111
add T5 encoding to text sets 111
declare qpl and qtm (qfonts) as aliases of ppl resp. ptm 112
Inheritance: add list for OT4 115
add list for T5 (requested by *Hàn Thế Thành*) 116
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR 122
settings for OT4 encoding (Computer Modern Roman, Palatino, Times) 118
settings for T5 encoded Computer Modern Roman 118
\DisableLigatures: new command: disable ligatures (requires pdfTeX 1.30) 84
\microtypecontext: new command: change setup context in the document 78
\MT@checklist@family: fix: add two missing \expandafters 48
\MT@define@option: fix: use true as the default value 97
\MT@detokenize@c: fix the non-e-TeX version 37
\MT@exp@two@n: new macros: less \expandafters 36
\MT@get@opt: new key 'preset' to set all characters to the specified value before loading the lists 53
\MT@is@active: redone: use \set@display@protect 71
\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning 70
\MT@maybe@do: redone 47
\MT@pdftex@no: compatibility with TeXLive hack (reported by *Herbert Voß*) 32
\MT@rem@from@clist: new macro: remove an item from a comma list 40
\MT@scale@factor: generalised 52
\MT@toks: use instead of \toks@ 34

Version 1.9a (2005/12/05)

General: new option: defersetup, by default true 98
remove superfluous test whether \pickup@font has changed 76
Documentation: add explanation for error message in DVI mode 24
add explanation for error message with non-Type 1 fonts 24
Font Sets: declare mdbch (mathdesign) as an alias of Charter 112
Protrusion: fix: remove _ from OT1 encoding 122
settings for T5 encoded Charter 118
\microtypesetup: inside the preamble, accepts all package options 102
\MT@check@font@cx: optimise context-sensitive setup 77
\MT@define@opt@keys: '⟨*file name*⟩/⟨*line number*⟩' as default list name 89
\MT@define@set@key@: don't expand variables immediately (requested by *Georg Verweyen*) 80
\MT@get@highlevel: no longer check whether defaults have changed 81
\MT@ifdefined@c@T: new macros: true case only 36
\MT@ifint: use \pdfmatch if available 37
\MT@ifstreq: use \pdfstrcmp if available 38
\MT@in@clist: fix 39
\MT@info@missing@char: info instead of warning (after *Michael Hoppe* reported that the 'fl' ligature is missing in Palatino SC) 52
\MT@is@feature: new macro: check for pdfTeX feature 42
\MT@map@clist@n: following LaTeX3 39
\MT@permute@@@@@: don't define permutations for unused encodings 94
\MT@rem@from@clist: fix 40
\MT@setup@: defer setup until the end of the preamble 42

Version 1.9b (2006/01/20)

General: compatibility with listings: sanitise more catcodes (reported by *Holger Uhr*) 45
compatibility with the extendedchar option of the listings package 45
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 42
\MT@ifdimen: use \pdfmatch if available 38
\MT@warn@code@too@large: fix calculation with present factor 52

Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by *Georg Verweyen*) 20

Protrusion: settings for URW Garamond 119

\MT@define@code@key@font: fix: context was ignored 88

\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in 1.9b) 88

Version 1.9d (2006/05/05)

Font Sets: md* instead of m series in basic sets 111

add QX encoding to text sets 111

Inheritance: add list for QX encoding (contributed by *Maciej Eder*) 116

Protrusion: settings for QX encoding (contributed by *Maciej Eder*) 124

settings for Euro symbols (Adobe, ITC, marvosym) 155

tweak AMS settings 147

\DeclareCharacterInheritance: fix: empty context 91

\MT@detokenize@n: new macro: use \detokenize if available 37

\MT@get@ex@opt: fix: evaluate preset 57

\MT@get@font@dimen: warning for zero fontdimen 52

\MT@get@opt: optimise: don't reset when preset option is set 53

set list name before presetting 53

\MT@is@active: support for Unicode (inputenc/utf8) 72

\MT@setupfont@hook: restore \% and \# when tex4ht is loaded (reported by *Peter Dyballa*) 44

\SetProtrusion: (et al.) optimise: unify keys for mandatory argument 86

(et al.) split keys of optional and mandatory argument 86

Version 1.9e (2006/07/28)

General: fix: default value for activate: true 97

no tracing with trace package 76

Documentation: add hint about unknown encodings 23

include LPPL 174

Font Sets: declare zeur and zeus (eulervm) as aliases of eur resp. eus (euler) 112

Inheritance: adapt to marvosym's changed encoding 117

Protrusion: complete settings for Euler Fraktur and Script fonts 154

fix: forgotten comma in mt-mvs.cfg; adapt to marvosym's changed encoding 155

settings for Euler Roman font 150

\DeclareCharacterInheritance: new key 'inputenc' to set the input encoding 91

\MT@rem@from@clist: model after \@removeelement 40

\MT@setup@: empty \MT@setup@ after use (compatibility with the combine class) 42

\SetExpansion: new key: inputenc 86

\SetProtrusion: new key: inputenc 86

Version 1.9f (2006/09/09)

Protrusion: fix: euler-vm did not load euler settings 151

\MT@curr@list@name: fix: \MessageBreak must not be expanded 73

\MT@gdef@n: new macros: global variants 36

\MT@glet: new macro 36

\MT@reset@context: only reset context if it has actually been changed 78

\MT@set@inh@list: fix: forgotten comma in the features list 92

\MT@set@inputenc: fix: input encoding must be set after the inheritance list has been parsed 54

\MT@set@named@keys: new macro: set name first, simplify parsing of optional argument 88

\SetProtrusion: (et al.) set catcodes before parsing optional argument 86

Version 2.0 (2007/01/14)

General: compatibility with listings: set catcode of backslash to zero (reported by *Steven Bath*) 45

compatibility with soul: register \textls and \lsstyle (beta:3) 45

maybe disable \MT@noligatures after the preamble 109

new option: babel, by default false (language-dependent setup suggested by *Ulrich Dirr*) (beta:1) 98

new option: letterspace, by default 100 (beta:1) 99

new package letterspace: a stripped-down version 1

option 'babel': disable Turkish babel's active characters (beta:8) 110

option 'babel': fix: switch off French babel's active characters with \shorthandoff instead of \NoAutoSpaceBeforeFDP (reported by *Daniel Flipo*) (beta:8) 110

option 'unit', \SetProtrusion: deprecate value

'relative' completely 90
option 'letterspacing' renamed to 'letterspace' 99
Documentation: add hint about how to increase font_max and font_mem_size 24
add hint about warning when tracking *and* expansion is applied to a font 24
add remark about 'draft' option disabling microtype (noted by *Michalis Miatidis*) 8
qualify hint about web documents with regard to older pdfTEX versions 22
qualify hints about expansion error messages with regard to older pdfTEX versions 24
Font Sets: add footnotesize and scriptsize sets 111
add smallcaps set 111
Kerning: remove letterspacing settings 159
\DeclareMicrotypeBabelHook: new command: interaction with babel (beta:1) 86
\lsstyle: fix: font switches don't pose a problem anymore (beta:8) 62
fix: letterspacing commands may be nested (beta:8) 62
new command: letterspacing (beta:1) 62
totally redone, using the new \letterspacefont 62
\MT@declare@sets: fix: empty size list when redefining set 80
\MT@is@symbol: made even more robust 72
\MT@pdftex@no: case 6: pdfTEX 1.40 (beta:1) 32
\MT@set@inputenc: sanitise catcodes before loading input encoding (problem with listings) . . 54
\MT@split@name: adjust to possible letterspacing . . 47
\SetExtraKerning: new command: additional kerning (beta:1) 87
\SetExtraSpacing: new command: adjustment of interword spacing (beta:1) 87
\SetTracking: new command: tracking 87
\textls: new command: letterspacing (beta:1) . . . 64
starred version: remove spaces around text . . . 64
\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations 31

Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition 75
compatibility with pinyin: disable microtype in \py@macron (reported by *Sven Naumann*) . . 45
fix: letterspace package forgot to load keyval 34
\lslig: new command: protect ligatures in letterspaced text 63
\MT@get@ls@basefont: redone: use \pdfmatch to make it bullet-proof 63
\textls: fix: use \hmode@bgroup 64

B Index

Links refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers refer to the code line where the corresponding entry is defined (underlined) resp. used.

C The LATEX Project Public License

LPPL Version 1.3c 2006-05-20

Preamble

The LATEX Project Public License (LPPL) is the primary license under which the 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
%   http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status `maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘`pig.dtx`’, ‘`pig.ins`’, and ‘`pig.sty`’ (the last being generated from ‘`pig.dtx`’ using ‘`pig.ins`’), the ‘Base Interpreter’ referring to any ‘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.