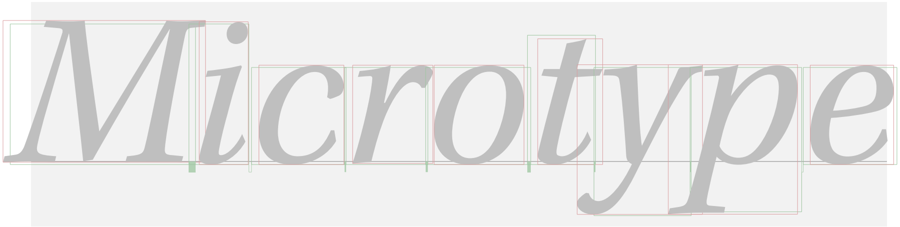

The `microtype` package

An interface to the micro-typographic extensions of pdfTEX

R Schlicht w.m.l@gmx.net

v1.9a — 2005/12/05

Abstract

The `microtype` package provides an interface to the micro-typographic extensions of pdfTEX: most prominently, character protrusion and font expansion, furthermore the possibility to disable all ligatures of a font.[1] It allows to apply these features to customizable 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.[2]

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 require pdfTEX 1.30. The package will by default enable the features that can safely be assumed to work.

---

[1] A preview of the next version with support for even more micro-typographical extensions is also included in this package. Footnote 11 on page 17 contains the details.

[2] Currently, this package provides settings for Computer Modern Roman, Palatino, Times, Adobe Garamond and Minion, Bitstream Charter, and the AMS math 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 two 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: *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Thế Thành's thesis:

> '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 [`pdfcprot`], which allowed LATEX users to employ character protrusion without having to mess much with the internals.

Font expansion, however, was still most difficult to utilize, since it required that the font metrics are available in all levels of expansion. Therefore, anybody who wanted to use 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 can be expected to prove 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.

Finally, the possibility to disable all ligatures of a font has been introduced. This may be useful when using typewriter fonts.

The `microtype` package provides an interface to all these micro-typographic extensions.[3] All micro-typographic aspects may be customized to your taste and needs in a straightforward manner. The next chapters will present a survey of all options and customization possibilities.

---

[3] Therefore, it is an alternative, not a supplement, to the `pdfcprot` package, which provides an interface to character protrusion.

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

`expansion` These are the main options to control the level of micro-typographic refinement, which the fonts in your document should gain. By default, the package is moderately greedy: Character protrusion will be enabled, font expansion will only be disabled in circumstances where 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 3.4).

Protrusion and expansion may be enabled or disabled independently from each `activate` 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 new pdfTeX):

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

```
\usepackage[protrusion,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.

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.

Whether ligatures should be disabled cannot be controlled via a package option but by using the `\DisableLigatures` command, which is explained in section 6.

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, p. 20). 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[4], 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.

---

[4] 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'.

`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:

```
\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 Miscellaneous Options

`DVIoutput` `true, false` `false`

pdfTEX is not only able to generate PDF output but can also spit out DVI files.[5] 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*. The `draft` and `final` options may also be inherited from the class options; of course, you can override them in the package options.

---

[5] TEX systems are beginning to switch to 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.

`config` ⟨*file name*⟩ `microtype`

Various settings for this package will be loaded from a main configuration file, by default `microtype.cfg` (see section 5.4). You can have a different configuration file loaded instead by specifying its name *without the extension*, e. g., `config=mycrotype`.

3.5 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). Using this command, you could for instance temporarily disable font expansion by saying:

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

4 Declaring Font Sets

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 expanded. You may want to customize the set of fonts that should get the benefit of micro-typographic treatment. This can be achieved by specifying attributes of the font that have to be matched for them to be taken into account.

`\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 set can subsequently be activated by calling:

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

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 = {T1,OT1,LY1,OT4,T5},
     family   = {rm*,sf*},
     series   = {m},
     size     = {normalsize,footnotesize,small,large}
   }
```

If you now call

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

in the document's preamble, only fonts in the text encodings T1, OT1, LY1, OT4 or T5, roman or sans serif families, normal (or 'medium') series, and in sizes called by `\normalsize`, `\footnotesize`, `\small` or `\large`, will be expanded. 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 = {T1,OT1,LY1,OT4,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 example above), it does not designate an NFSS code, but will expand to the document's `\`⟨*value*⟩`default`, e. g. `\rmdefault`. For example, if you want to include the bold font, too, you should say '`bf*`' instead of '`b`' (`\bfdefault` for Computer Modern is '`bx`', while for other fonts, it might be '`b`' or even '`sb`'). A single asterisk means `\`⟨*characteristic*⟩`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 characteristics, 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

Table 1: Predefined font sets

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

`alltext`
`(allmath)`
	OT1, T1, LY1,
	OT4, T5, TS1
	(OML, OMS, U)
		–	–	–	–

`basictext`
`(basicmath)`
	OT1, T1, LY1,
	OT4, T5
	(OML, OMS)
		`\rm*`,
		`\sf*`
			m	–	`\normalsize`,
					`\footnotesize`,
					`\small`, `\large`

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

'*' = '`default`'

sized, possibly italic, typewriter font (in contrast to, say, the small one), this is how you could declare the set:

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

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

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

```
\usepackage[protrusion=allmath,expansion=basicmath]{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 '`basictext`' font set will be used for font expansion, the '`alltext`' set for character protrusion.

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 amount of protrusion. 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 customizing these finer aspects of micro-typography.

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 characters may be entered directly or in the LaTeX 7-bit way of defining them: both `Ä` and `\"A` are valid, provided the character is actually included in the encoding(s). You also have the possibility to declare lists of characters that should inherit protrusion or expansion factors (see section 5.3).

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.

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`. 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.[6] The encoding must always match.

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 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`.[7]

---

[6] For the interested, table 3 on page 59 presents the exact order.

[7] 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.

`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`.

`context` The scope of the list may be limited to a certain context. For an example application, see section 7.

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.

The set of fonts is declared in the same way as for `\SetProtrusion`.

Options:

`name`, `load`, `preset`, `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, 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 printed in 10pt:[8]

```
\SetExpansion
   [ stretch = 30,
     shrink  = 60 ]
   { encoding = *,
     size = 10.001 }
   { }
\newcommand{\expandpar}[1]
   {{\fontsize{10.001}{\baselineskip}\selectfont #1}}
% ...
\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 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.

---

[8] 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.

5.3 Character Inheritance

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

In most cases, accented characters should inherit the protrusion resp. expansion factors 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 lists of characters, so that you then only have to set up the base characters. With the optional argument, which may contain a comma-separated list of features, you may confine the scope of the list. The font set can be declared in the usual way, with the only exception that you must specify exactly one encoding. The inheritance lists are to be 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.

5.4 Configuration Files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion and expansion 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 3.4).

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, Adobe Garamond and Minion[9], for Bitstream Charter and the AMS math fonts. Table 2 lists them all.

If you have created a file for another font and you are willing to share, don't hesitate to send it to me so that it can be included in future releases of this package.

`\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

---

[9] 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, (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
Adobe Minion (`pmnx`, `pmnj`)	OT1, T1, LY1, TS1	n, it, (sl)[d], sc, si
Palatino (`ppl`, `pplx`, `pplj`)[e]	OT1, OT4, T1, LY1, (TS1)[a]	n, it, (sl)[d], sc
Times (`ptm`, `ptmx`, `ptmj`)[f]	OT1, OT4, T1, LY1, (TS1)[a]	n, it, (sl)[d], sc
Computer Modern math fonts (`cmsy`, `cmm`)	OML/OMS	n/it
AMS math fonts (`msa`, `msb`, `euf`, `eus`)	U	n

[a] Incomplete
[b] Also used for: Latin Modern (`lmr`), `ae` (`aer`), `zefonts` (`zer`), `eco` (`cmor`), `hfoldsty` (`hfor`)
[c] Also used for: `mathdesign`/Charter (`mdbch`)
[d] Settings inherited from italic shape
[e] Also used for: `pxfonts` (`pxr`), `qfonts`/QuasiPalatino (`qpl`)
[f] Also used for: `txfonts` (`txr`), `qfonts`/QuasiTimes (`qtm`)

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 in the default configuration file, along with others for the virtual fonts provided by the packages `ae`, `zefonts`, `eco` and `hfoldsty`.

`\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.[10] This command will load the file `mt-`⟨*font name*⟩`.cfg`.

6 Disabling Ligatures

`\DisableLigatures` {⟨*set of fonts*⟩}

A new feature has been introduced with pdfTEX 1.30: The possibility to completely disable all ligatures of a font (which will also switch off kerning). While this purposely *lowers* the micro-typographic quality instead of raising it, it is especially

---

[10] Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 13.7.2.

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* }
```

7 Context-sensitive Setup

In previous versions of `microtype`, each font was set up exactly once for the entire document. Since version 1.9, it is possible to apply different settings to a font depending on the context it appears in.

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

This command may be used anywhere in the document (also in the preamble) to change the micro-typographic context. For each feature (`protrusion`, `expansion`), one context may be specified. Only settings which have been specified with the corresponding '`context`' keyword will then be applied. This makes it possible to use different settings for different parts of the document.[11]

8 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

---

[11] This feature is especially useful for the new experimental extensions of pdfTEX: adjustment of interword spacing (glue) and the possibility to specify additional character kerning. The former may improve the appearance of the text even more, the latter allows for instance to insert small spaces before certain characters (e. g., for typesetting in the French tradition) without having to use active characters; also, letterspacing can be implemented in a robust way. Currently, these extensions are only available through patches from `http://pdftex.sarovar.org/`. However, if you are adventurous, know how to apply the patches and you are able to compile pdfTEX yourself, you can easily experiment with them, since `microtype` already supports these new extensions. To generate the extended version of the `microtype` package and its documentation, simply remove the comments before '`\betatrue`' near the beginning of `microtype.ins` and `microtype.dtx`.

`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. Because each expanded instance of the font will be embedded in the PDF file, 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.

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 aware of the following issues:

- When using 8-bit characters in the configuration, `inputenc` must be loaded first. Unicode input in the configuration is currently not supported.
- The `CJK` package, like `microtype`, hooks into the LaTeX font selection scheme. Therefore, both packages probably don't cooperate well. However, since I know nothing about `CJK`, I would appreciate feedback on the interaction of both packages – be it positive or negative.

You might want to disable protrusion in `verbatim` *environments.* As you know by now, `microtype` will by default apply character protrusion to all fonts part of 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 of 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 incantation, added to the document's preamble, would serve the same purpose:[12]

```
\makeatletter
\g@addto@macro\verbatim{\pdfprotrudechars=0 \pdfadjustspacing=0\relax}
\makeatother
```

---

[12] 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:

- `Warning: pdflatex: font ptmr8r cannot be expanded (not an included Type1 font)`

  Font expansion can only be applied if the font is actually embedded in the PDF file. If you receive 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`. Otherwise, consult the local guide of your TEX system.
- `Warning: pdflatex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found`

  Furthermore, automatic font expansion requires Type 1 fonts. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap font. This is not possible, unless you manually create expanded instances of the fonts.
- `! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.`

  Such an error message could 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.
- `! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].`

  When applying micro-typographic enhancement to a large document with a lot of fonts, you may be running out of pdfTEX memory. The memory can be increased by setting `pdf_mem_size` to a larger value (maximum 524 288). For teTEX-based systems, change the settings in `texmf.cnf`, for MiKTEX, in the file `miktex.ini`. Beginning with version 1.30 of pdfTEX, memory will grow dynamically, so that this problem can no longer arise.

9 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`.[13]

---

[13] Should you have lots of `pdfcprot` configuration files lying around, I can also provide you with a TEX conversion script. Just ask me.

10 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].

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.

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 or helped otherwise (in chronological order): *Ulrich Dirr*, *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* and *Ralf Stubner*.

11 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, *The pdfTEX user manual*, August 21, 2005. (Available from CTAN at `/systems/pdftex/manual`; latest version at `http://sarovar.org/projects/pdftex/`)

LATEX3 Project Team, *LATEX 2ε font selection*, February 10, 2004. (Available from CTAN at `/macros/latex/doc/fntguide.pdf`)

Carsten Schurig, *The `pdfcprot.sty` package*, August 14, 2002. (Available from CTAN at `/macros/latex/contrib/pdfcprot/`)

12 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 fixes are swept under the rug.

1.9a (5. 12. 2005)

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

Protrusion settings for `T5` encoded Charter

1.9 (28. 10. 2005)

New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later; see section 6)

New command `\microtypecontext` to change the configuration context; new key ‘`context`’ for the configuration commands (see section 7)

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 (23. 6. 2005)

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 3.4)

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 files 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 (23. 3. 2005)

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 5.4)
Hook `\Microtype@Hook` for font package authors (see section 13.7.2)
New option '`verbose=errors`' to turn all warnings into errors
Disable expansion inside `\showhyphens`
Warning when running in draft mode

1.6a (2. 2. 2005)
Compatibility with the `frenchpro` package

1.6 (24. 1. 2005)
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 (15. 12. 2004)
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
Compatibility with Turkish `babel`

1.4b (26. 11. 2004)
`\UseMicrotypeSet` requires the set to be declared (see section 4)

1.4 (12. 11. 2004)
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 3.5)
New option: '`final`'

1.3 (27. 10. 2004)
Compatibility with the `german` and `ngerman` packages

1.2 (3. 10. 2004)
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 5.4)
More tests for sanity of settings and whether all fonts will be set up
More robust parsing of sizes in font sets

1.1 (21. 9. 2004)
Protrusion settings for Adobe Minion, contributed by Harald Harders
New command: `\DeclareCharacterInheritance` (see section 5.3)
Characters may also be specified as octal or hexadecimal numbers (see section 5)
Configuration file names in lowercase (see section 5.4)

1.0 (11. 9. 2004)
First CTAN release

13 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.
- `config`: Surrounds all configuration modules.
  - `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.
  - `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 AMS Euler Fraktur font (`mt-euf.cfg`).
    - `eus`: Settings for AMS Euler script font (`mt-eus.cfg`).
- `test`: A helper file that may be used to create and test protrusion settings (`test-microtype.tex`).
- `beta`: Support for features not yet included in an official release of pdfTEX.

And now for something completely different.

```
⟨*package⟩
```

These are all commands for the outside world. We define them here as dummy 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*\DeclareMicrotypeAlias[2]{}
\newcommand*\SetProtrusion[3][]{}
\newcommand*\SetExpansion[3][]{}
\newcommand*\DisableLigatures[1]{}
\newcommand*\DeclareCharacterInheritance[3][]{}
\newcommand*\LoadMicrotypeFile[1]{}
\newcommand*\microtypesetup[1]{}
\newcommand*\microtypecontext[1]{}
⟨*beta⟩
\newcommand*\SetExtraSpacing[3][]{}
```

```
\newcommand*\SetExtraKerning[3][]{}
\newcommand*\DeclareMicrotypeBabelHook[2]{}
\newcommand\textls[2][]{#2}
\newcommand\lsstyle{}
⟨/beta⟩
```

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}
⟨beta⟩\@onlypreamble{\DeclareMicrotypeBabelHook}
```

`\MT@old@cmd`

The old command names had one more hunch.

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

```
\newcommand*\DeclareMicroTypeSet{%
 \MT@old@cmd\DeclareMicroTypeSet
            \DeclareMicrotypeSet}
\newcommand*\UseMicroTypeSet{%
 \MT@old@cmd\UseMicroTypeSet
            \UseMicrotypeSet}
\newcommand*\DeclareMicroTypeAlias{%
 \MT@old@cmd\DeclareMicroTypeAlias
            \DeclareMicrotypeAlias}
\newcommand*\LoadMicroTypeFile{%
 \MT@old@cmd\LoadMicroTypeFile
            \LoadMicrotypeFile}
```

`\MT@MT`

This is us.

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

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

Communicate.

```
\def\MT@error{\PackageError{\MT@MT}}
\def\MT@warning{\PackageWarning{\MT@MT}}
\def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
\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,dvi) to find out what went wrong.}}
\def\MT@info{\PackageInfo{\MT@MT}}
\def\MT@info@nl#1{\MT@info{#1\@gobble}}
⟨!debug⟩\let\MT@vinfo\@gobble
```

Debug. Cases for `\tracingmicrotype`:

0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```
⟨*debug⟩
\let\MT@vinfo\MT@info@nl
\newcount\tracingmicrotype
\tracingmicrotype=\tw@
\def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1\relax\else\MT@info{#2}\fi}
\def\MT@dinfo@nl#1#2{\ifnum\tracingmicrotype<#1\relax\else\MT@info@nl{#2}\fi}
⟨/debug⟩
```

13.1 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, there are 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 ($\geq$ 0.14h)

4: + automatic font expansion; default `\efcode` = 1000 ($\geq$ 1.20)

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

6: + adjustment of interword spacing; extra kerning; `\pdfmatch`[14] ($\geq$ 1.4x)

```
\let\MT@pdftex@no\z@
```

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 defined `\pdftexversion` to `\relax`.

```
\ifx\pdftexversion\@undefined \else
  \ifx\pdftexversion\relax \else
⟨debug⟩\MT@dinfo@nl{0}{running pdftex \the\pdftexversion(\pdftexrevision)}
⟨beta⟩    \ifx\pdfadjustinterwordglue\@undefined
      \def\MT@pdftex@no{5}
⟨beta⟩    \else \def\MT@pdftex@no{6}\fi
    \ifnum\pdftexversion < 130
      \def\MT@pdftex@no{4}
      \ifnum\pdftexversion < 120
        \let\MT@pdftex@no\thr@@
```

---

[14] This command was actually introduced in 1.30, however, pdfTEX ran in a buffer overflow with strings larger than 1024 bytes.

```
    \ifnum\pdftexversion = 14
     \ifnum \expandafter'\pdftexrevision < 'h
      \let\MT@pdftex@no\tw@
      \ifnum \expandafter'\pdftexrevision < 'f
       \let\MT@pdftex@no\@ne
      \fi
     \fi
    \else
     \ifnum\pdftexversion < 14
      \let\MT@pdftex@no\@ne
     \fi
    \fi
   \fi
  \fi
 \fi
\fi
⟨debug⟩\MT@dinfo@nl{0}{pdftex no: \number\MT@pdftex@no}
```

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

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

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

```
\ifnum\MT@pdftex@no<\tw@
 \AtEndOfPackage{\let\@unprocessedoptions\relax}%
 \let\CurrentOption\@empty
 \MT@warning@nl{%
  \ifcase\MT@pdftex@no
   You don't seem to be using pdftex.\MessageBreak
  \or
   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
  \fi
  All micro-typographic features will be disabled}
 \expandafter
 \endinput
\fi
```

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\"%
}
```

\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.

```
\RequirePackage{keyval}[1997/11/10]
```

\MT@toks

We need a token register.

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

\ifMT@protrusion
\ifMT@expansion
\ifMT@auto
\ifMT@selected
\ifMT@noligatures
\ifMT@draft
\ifMT@spacing
\ifMT@kerning
\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
⟨*beta⟩
\newif\ifMT@spacing
\newif\ifMT@kerning
\newif\ifMT@babel
⟨/beta⟩
```

\MT@ifdocument

Our private test whether we're still in the preamble

```
\newif\ifMT@document
```

\MT@pr@level
\MT@pr@factor
\MT@pr@unit
\MT@ex@level
\MT@ex@factor
\MT@stretch
\MT@shrink
\MT@step
\MT@sp@factor
\MT@sp@unit
\MT@kn@factor
\MT@kn@unit
\MT@letterspacing

… and numbers.

```
\let\MT@pr@level\tw@
\let\MT@pr@factor\@m
\let\MT@pr@unit\@empty
\let\MT@ex@level\tw@
\let\MT@ex@factor\@m
\let\MT@stretch\m@ne
\let\MT@shrink \m@ne
\let\MT@step   \m@ne
⟨*beta⟩
\let\MT@sp@factor\@m
\let\MT@kn@factor\@m
```

Default unit for spacing settings is `space`, default unit for kerning is 1em.

```
\let\MT@sp@unit\m@ne
\def\MT@kn@unit{1em}
\let\MT@letterspacing\m@ne
⟨/beta⟩
```

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

\MT@factor@default \MT@stretch@default \MT@shrink@default \MT@step@default

Default values for expansion.

```
\def\MT@factor@default{1000 }
\def\MT@stretch@default{20 }
\def\MT@shrink@default{20 }
\def\MT@step@default{4 }
```

\MT@letterspacing@default

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

```
⟨beta⟩\def\MT@letterspacing@default{100 }
```

13.2 Compatibility

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

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent`
- `\showhyphens`

\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
}
\@ifpackageloaded{pdfcprot}\MT@pdfcprot@error\relax
```

\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).

```
\def\MT@ledmac@setup{%
  \ifMT@protrusion
    \MT@requires@pdftex5{%
```

```
      \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}%
      }%
    }{%
      \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@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{%
```

The `chemsym` package redefines, among other commands, the Hungarian umlaut `\H` in a way that cannot be parsed by `microtype`. As a work-around, we restore the usual definition of `\H` before setting up the font (which will be done inside a group). — Since version 1.7, this is no longer needed, since our character parsing is robust enough now.

Same for `statex`. — No longer needed, either.

Spanish `babel` modifies the percent character, storing the original meaning in `\percentsign`.

```
  \@ifpackagewith{babel}{spanish}{%
    \MT@ifdefined@c@T\percentsign{\let\%\percentsign}%
  }\relax
```

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.)

```
  \@ifpackageloaded{csquotes}{%
    \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax
  }\relax
```

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

```
  \@ifpackageloaded{hyperref}{%
    \chardef\%'\%
    \chardef\#'\#
  }\relax
}
```

`\MT@setup@microtype` 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`.

```
\let\MT@setup@microtype\@empty
```

`\MT@addto@setup` We use our private hook to have better control over the timing.

```
\def\MT@addto@setup#1{%
   \begingroup
      \MT@toks\expandafter{\MT@setup@microtype#1}%
      \xdef\MT@setup@microtype{\the\MT@toks}%
   \endgroup
}
```

It will be executed at the end of the preamble.

```
\AtBeginDocument{\MT@setup@microtype}
```

Check again at the end of the preamble.

```
\MT@addto@setup{%
  \@ifpackageloaded{pdfcprot}\MT@pdfcprot@error\relax
  \@ifpackageloaded{ledmac}\MT@ledmac@setup\relax
```

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

```
  \let\MT@setupfont@hook\@empty
  \@ifpackagewith{babel}{spanish}{%
    \g@addto@macro\MT@setupfont@hook{%
      \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
  }\relax
  \@ifpackageloaded{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}%
    }%
  }\relax
  \@ifpackageloaded{hyperref}{%
    \g@addto@macro\MT@setupfont@hook{%
      \chardef\%'\%
      \chardef\#'\#
    }%
```

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

```
    \pdfstringdefDisableCommands{%
      \let\pickup@font\MT@orig@pickupfont
⟨*beta⟩
      \let\lsstyle\@empty
      \def\textls#1#{\@firstofone}%
⟨/beta⟩
    }%
  }\relax
}
```

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

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

13.3 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
```

`\MT@def@n` This is `\@namedef`.

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

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

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

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

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

`\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@exp@string` Remove trailing space.

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

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

```
\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}
```

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

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{%
⟨debug⟩\MT@dinfo@nl{0}{running etex}%
  \def\MT@ifdefined@c@TF#1{%
    \ifdefined#1%
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
  \def\MT@ifdefined@c@T#1{%
    \ifdefined#1%
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
  }
  \def\MT@ifdefined@n@TF#1{%
    \ifcsname#1\endcsname
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
  \def\MT@ifdefined@n@T#1{%
    \ifcsname#1\endcsname
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
  }
}{%
⟨debug⟩\MT@dinfo@nl{0}{not running etex}%
  \def\MT@ifdefined@c@TF#1{%
    \ifx#1\@undefined
      \expandafter\@secondoftwo
    \else
      \expandafter\@firstoftwo
    \fi
  }
  \def\MT@ifdefined@c@T#1{%
    \ifx#1\@undefined
      \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
  }
  \def\MT@ifdefined@n@T#1{%
    \begingroup\MT@exp@two@c\endgroup
    \ifx\csname #1\endcsname\relax
      \expandafter\@gobble
    \else
      \expandafter\@firstofone
    \fi
  }
}
```

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

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

`\MT@ifnumber` Test whether argument is a number [0-9] (using an old trick by Mr. Arseneau).

```
\MT@requires@pdftex5{%
  \def\MT@ifnumber#1{%
    \ifcase\pdfmatch{^[0-9]+ *$}{#1}\relax
      \expandafter\@secondoftwo
    \else
      \expandafter\@firstoftwo
    \fi
  }
}{%
  \def\MT@ifnumber#1{%
    \if!\ifnum9<1#1!\else?\fi
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
}
```

`\MT@ifdimen` Test whether argument is dimension (or number).

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

`\MT@ifgt` `\MT@iflt` `\MT@ifeq` Test whether dimensions are smaller, larger or equal.

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

```
\def\MT@iflt#1#2{%
  \ifdim #1\p@ < #2\p@
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
\def\MT@ifeq#1#2{%
  \ifdim #1\p@ = #2\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.

```
\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{}
```

`\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*⟩. Using `\pdfmatch` and `\pdflastmatch` here would be really slow!

```
\def\MT@rem@from@clist#1#2{%
  \def\x##1,#1,##2##3\@nnil{%
    \ifx##2\@empty\else
      \ifx\\##1\\\global\let#2\@gobble
        \else\xdef#2{\@gobble##1}\fi
      \def\x####1,#1,####2####3\@nnil{%
        \xdef#2{#2####1}%
      }%
      \x,##2##3,#1,\@nnil
    \fi
  }%
  \expandafter\x\expandafter,#2,#1,\@empty\@nnil
}
```

`\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@ifeq{#2}\m@ne{%
    \MT@ifeq{#1}\MT@size
      \MT@inlist@true
      \relax
    }{%
    \MT@iflt\MT@size{#1}\relax{%
      \MT@iflt\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@do@font`

Execute ⟨*#1*⟩ 255 times.

```
\def\MT@do@font#1{%
  \@tempcnta\z@
  \MT@loop #1%
    \advance\@tempcnta \@ne
    \ifnum\@tempcnta < \@cclvi \MT@repeat
}
```

`\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`

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}
⟨*beta⟩
\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}
⟨/beta⟩
```

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

These we also need the other way round.

```
\def\MT@rbba@protrusion{pr}
\def\MT@rbba@expansion{ex}
⟨*beta⟩
\def\MT@rbba@spacing{sp}
\def\MT@rbba@kerning{kn}
⟨/beta⟩
```

13.4 Setting up a font

`\MT@setupfont`

Setting up a font entails checking whether protrusion/expansion is desired for the current font (`\MT@font@name`), and if so, adjusting `\lpcode` and `\rpcode` (protrusion) and `\efcode` (expansion) for each character.

```
\def\MT@setupfont{%
  \ifx\MT@vinfo\MT@info@nl
    \MT@info{Setting up font '\MT@exp@string\MT@font'}\fi
```

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

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

The font properties must be extracted from `\MT@font@name`, 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
```

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.

```
⟨*beta⟩
  \MT@spacing
  \MT@kerning
⟨/beta⟩
```

Disable ligatures?

```
  \MT@noligatures
}
```

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

Split up the font name.

```
\def\MT@split@name#1/#2/#3/#4/#5\@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 characteristic is in the list.

```
    \expandafter\expandafter\expandafter
      \MT@in@clist\expandafter\expandafter\expandafter
        {\csname MT@#1\expandafter\endcsname\expandafter}%
         \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 characteristic has not been defined at all, the font should be expanded resp. protruded.

```
⟨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 `\expandafter`s 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}}%
}
```

13.4.1 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{%
  \MT@reset@pr@codes
```

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

```
  \MT@if@list@exists{%
    \MT@get@dimen@six
    \MT@get@opt
```

Get the name of the inheritance list and parse it.

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

Load additional lists?

```
    \MT@load@list{\MT@pr@c@name}%
```

Load the main list.

```
    \edef\MT@curr@list@name{protrusion list '\MT@pr@c@name'}%
    \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
    \expandafter\MT@pr@do\@tempc,\relax,%
```

```
  }\relax
}
```

`\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@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@dimen@six{%
  \ifnum\fontdimen6\MT@font=\z@
    \MT@warning@nl{%
      Font '\MT@exp@string\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@pdftex\thr@@{%
  \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=\number\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` (neutralized).

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

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

```
\def\MT@get@font@dimen#1{%
  \MT@count=\number\fontdimen#1\MT@font
}
```

`\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@exp@string\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@exp@string\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
    \@tempcnta=\csname MT@\MT@feat @max\endcsname
    \MT@warn@code@too@large
  \else
    \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
      \@tempcnta=\csname MT@\MT@feat @min\endcsname
      \MT@warn@code@too@large
    \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 file.

```
\def\MT@warn@code@too@large{%
  \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=\@tempcnta
}
```

`\MT@get@opt`

The optional argument to `\SetProtrusion`, `\SetExtraSpacing` and `\SetExtraKerning` (`\SetExpansion` is being dealt with in `\MT@get@ex@opt`).

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

`\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?

```
  \MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
    \csname MT@preset@\MT@feat\endcsname
  }%
}
```

`\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=\number\@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@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 width 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}%
}
```

13.4.2 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@load@list{\MT@ex@c@name}%
    \edef\MT@curr@list@name{expansion list '\MT@ex@c@name'}%
    \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@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@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @stretch}{%
    \MT@let@cn\MT@stretch@{MT@ex@c@\MT@ex@c@name @stretch}%
    \MT@vinfo{... : Setting stretch limit to \number\MT@stretch@}%
  }{%
    \let\MT@stretch@\MT@stretch
  }%
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @shrink}{%
    \MT@let@cn\MT@shrink@{MT@ex@c@\MT@ex@c@name @shrink}%
    \MT@vinfo{... : Setting shrink limit to \number\MT@shrink@}%
  }{%
    \let\MT@shrink@\MT@shrink
  }%
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @step}{%
    \MT@let@cn\MT@step@{MT@ex@c@\MT@ex@c@name @step}%
    \MT@vinfo{... : Setting expansion step to \number\MT@step@}%
  }{%
    \let\MT@step@\MT@step
  }%
  \MT@ifdefined@n@TF{MT@ex@c@\MT@ex@c@name @auto}{%
```

```
    \MT@let@cn\MT@auto@{MT@ex@c@\MT@ex@c@name @auto}%
    \def\@tempa{autoexpand}%
    \MT@vinfo{... : \ifx\@tempa\MT@auto@ En\else Dis\fi
                    abling automatic expansion}%
  }{%
    \let\MT@auto@\MT@auto
  }%
}
```

`\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
}
```

```
⟨*beta⟩
```

13.4.3 Interword Space (Glue)

`\MT@spacing` Adjustment of interword spacing? Only for sufficiently new versions of pdfTEX.

```
\MT@requires@pdftex6{%
  \def\MT@spacing{\MT@maybe@do{sp}}
}{%
  \let\MT@spacing\relax
}
```

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

```
\def\MT@set@sp@codes{%
  \MT@reset@sp@codes
  \MT@if@list@exists{%
    \MT@get@dimen@six
    \MT@get@opt
    \MT@get@inh@list
    \MT@load@list{\MT@sp@c@name}%
    \edef\MT@curr@list@name{spacing list '\MT@sp@c@name'}%
    \MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}%
    \expandafter\MT@sp@do\@tempc,\relax,%
  }\relax
}
```

`\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
}
```

13.4.4 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}}
}{%
  \let\MT@kerning\relax
}
```

`\MT@set@kn@codes`

```
\def\MT@set@kn@codes{%
  \MT@reset@kn@codes
  \MT@if@list@exists{%
    \MT@get@dimen@six
    \MT@get@opt
    \MT@get@inh@list
    \MT@load@list{\MT@kn@c@name}%
    \edef\MT@curr@list@name{kerning list '\MT@kn@c@name'}%
    \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
    \expandafter\MT@kn@do\@tempc,\relax,%
  }\relax
}
```

\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
```

Letterspacing factor.

```
  \MT@ifstreq\MT@kn@context{letterspacing}{%
    \@tempcnta\MT@kn@factor@\relax
    \MT@scale\@tempcnta \MT@letterspacing@ \@m
    \edef\MT@kn@factor@{\number\@tempcnta}%
  }\relax
  \MT@preset@aux{#1}\@tempa
  \MT@preset@aux{#2}\@tempb
  \MT@set@all@kn\@tempa\@tempb
}
```

13.4.5 Letterspacing

`\lsstyle` Letterspacing is a special case of extra kerning. It will temporarily switch kerning on, activate the `all` font set and load the `letterspacing` context. The list must have the name `letterspacing`, so that the factor will be applied.

```
\MT@requires@pdftex6{%
  \DeclareRobustCommand\lsstyle{%
    \ifMT@kerning
```

We have to add the current font to the active kerning font set, so that it is guaranteed to be reset. This will fail for font switches inside `\lsstyle`.

```
      \begingroup
      \escapechar\m@ne
      \expandafter\MT@exp@two@n\expandafter\MT@in@clist
          \csname\curr@fontshape/\f@size\expandafter\endcsname
          \csname MT@knlist@font@\MT@kn@setname\endcsname
      \ifMT@inlist@ \else
        \expandafter\MT@xadd
          \csname MT@knlist@font@\MT@kn@setname\endcsname
          {\MT@exp@string\csname\curr@fontshape/\f@size\endcsname,}%
      \fi
      \endgroup
    \fi
    \MT@kerningtrue
    \pdfappendkern\@ne
    \pdfprependkern\@ne
    \def\MT@kn@setname{all}%
    \MT@ifdefined@c@TF\MT@letterspacing@\relax{%
      \let\MT@letterspacing@\MT@letterspacing
    }%
    \microtypecontext{kerning=letterspacing}%
  }
}{%
  \DeclareRobustCommand\lsstyle{%
    \MT@warning{Letterspacing only works with pdftex version 1.4x\MessageBreak
      or newer. You might want to use the 'soul' package\MessageBreak
      instead}%
    \global\let\lsstyle\relax
  }
}
```

`\textls`
`\MT@letterspacing@`

This command may be used like the other text commands. The optional argument may be used to change the letterspacing factor.

```
\DeclareRobustCommand\textls[2][]{%
  \MT@ifempty{#1}{%
    \let\MT@letterspacing@\@undefined
  }{%
    \KV@@sp@def\MT@letterspacing@{#1 }%
  }%
  {\lsstyle #2}%
}
⟨/beta⟩
```

13.4.6 Disabling Ligatures

`\MT@noligatures`

The possibility to disable ligatures is a new features of pdfTEX 1.30.0.

```
\MT@requires@pdftex5{%
  \def\MT@noligatures{%
    \csname ifMT@\MT@abbr@nl\endcsname
      \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}%
      }%
    \else
      \MT@dofalse
    \fi
    \ifMT@do
      \pdfnoligatures\MT@font
      \MT@vinfo{... Disabling ligatures}%
    \fi
  }
}{%
  \let\MT@noligatures\relax
}
```

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

```
\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@begin@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.

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

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

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%
```

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	●	-	●	-	●	-	●	-	●	-	●	-	●	-	●	-

```
  \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@ex@c@name')}%
    \else
      \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list
                    '\@nameuse{MT@\MT@feat @c@name}'}%
    \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}%
    \else
      \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
        for font\MessageBreak'\MT@exp@string\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
                    '\@nameuse{MT@\MT@feat @inh@name}'}%
⟨/debug⟩
    \MT@let@cn\@tempc{MT@\MT@feat @inh@\csname MT@\MT@feat @inh@name\endcsname}%
```

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

```
    \ifx\@tempc\@empty \else
⟨debug⟩\MT@dinfo@nl{1}{parsing inheritance list ...}%
      \MT@let@cn\MT@inh@name{MT@\MT@feat @inh@name}%
      \def\MT@curr@list@name{inheritance list\MessageBreak'\MT@inh@name'}%
      \expandafter\MT@inh@do\@tempc,\relax,%
      \global\MT@let@nc{MT@\MT@feat @inh@\csname MT@\MT@feat @inh@name\endcsname}\@empty
    \fi
  }{%
    \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
  }%
}
```

13.4.8 Translating Characters into Slots

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

`\MT@get@slot` `\MT@char` `\Mt@char` 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.

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\expandafter\expandafter{\expandafter\string\@tempa}%
  \edef\MT@char{\expandafter\meaning\@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`.

```
        \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.

```
        \MT@ifdefined@n@TF{\MT@encoding\MT@detokenize\@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.

```
          \MT@exp@two@c\MT@is@char\MT@char\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@unknown@i
      \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 1\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@ifnumber{#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.

Previous solution, slightly more robust:

```
\def\MT@is@active#1#2\@nil{%
   \ifx\\#2\\%
      \ifnum\catcode'#1 = \active
         \toks@=\expandafter\expandafter\expandafter{\@tempa}%
         \expandafter\MT@active@inpenc\the\toks@\relax\relax
         \edef\@tempa{\the\toks@}%
         \edef\x{\MT@toks={\the\MT@toks\space(= \the\toks@)}}\x
      \fi
   \fi
}
\def\MT@active@inpenc#1#2#3\relax{%
   \ifx#1\IeC
      \def\IeC##1{\toks@={##1}}%
      \the\toks@
      \expandafter\MT@active@inpenc\the\toks@\relax\relax
   \fi
   \ifx#1\@tabacckludge
      \def\@tabacckludge##1##2{%
         \toks@=\expandafter{\csname\string##1\endcsname#3}}%
      \the\toks@
   \fi
   \ifx#1\@inpenc@undefined@
      \def\@inpenc@undefined@##1{%
         \edef\x{\toks@={%
            undefined^^J(\MT@MT)\@spaces\@spaces\@spaces\@spaces
            in input encoding ''##1''}}\x}%
      \the\toks@
   \fi
}
```

Unfortunately, the `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 currently not supported.

```
\def\MT@is@active#1#2\@nil{%
  \ifx\\#2\\%
    \ifnum\catcode'#1 = \active
      \begingroup
        \set@display@protect
        \def\IeC##1{##1}%
```

The character is undefined in the encoding.

```
        \def\@inpenc@undefined@##1{undefined^^J%
          (\MT@MT)\@spaces\@spaces\@spaces\@spaces
          in input encoding ''##1''}%
        \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
  \fi
}
```

`\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{%
  \edef\@tempa{\expandafter
               \csname\expandafter
                 \MT@encoding\expandafter
                 \string\@tempa
               \endcsname}%
  \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
      \meaning\expandafter\@tempa\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\@tempa\relax\relax
  \fi
}
```

`\MT@is@char`
`\MT@charstring`

Here we define a helper macro that inspect the `\meaning` of its argument.

```
⟨debug⟩\def\Mt@info{\MT@dinfo@nl{3}}
\begingroup
  \catcode'\/=0
  /MT@map@tlist@n{/\/C/H/A/R}/@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@info{> '/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.

```
    \edef\@tempa{\expandafter
                   \csname\expandafter
                     \string\csname\MT@encoding\endcsname
                     \string#1-%
                     \string#2%
                   \endcsname}%
    \expandafter\MT@is@letter\@tempa\relax\relax
  \fi
}
```

`\MT@detokenize` Translate a macro into a token list. We must be cautious not to stumble over accented characters consisting of two commands, like `\'\i` or `\U\CYRI`. 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.

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

Some warning messages, for performance reasons separated here.

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

```
\let\MT@curr@list@name\@empty
```

`\Mt@warn@ascii` For characters with character code > 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@unknown@i` Not all of the string has been parsed.

```
\def\MT@warn@unknown@i{%
  \MT@warning@nl{%
    Unknown slot number of character '\the\MT@toks' in\MessageBreak
    font encoding '\MT@encoding'. Make sure it's a single\MessageBreak
    character (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 '\the\MT@toks' in\MessageBreak
    font encoding '\MT@encoding' in \MT@curr@list@name}%
}
```

13.4.9 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 favorite class, the `memoir` class, 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
\MT@addto@setup{%
  \g@addto@macro\do@subst@correction{%
    \xdef\MT@font{\csname \curr@fontshape/\f@size\endcsname}%
  }
```

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

```
  \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.\MessageBreak
      Double-check whether micro-typography is indeed\MessageBreak
      applied to the document.\MessageBreak (Hint: Turn on 'verbose' mode)%
    }
  \fi
```

Then we append our stuff.

```
  \g@addto@macro\pickup@font{%
    \begingroup
      \escapechar\m@ne
```

If `\MT@font@name` 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 one-level 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
          \MT@register@subst@font
        \fi
      \fi
      \MT@check@font
      \ifMT@inlist@ \else
        \MT@setupfont
        \MT@register@font
      \fi
    \endgroup
    \global\let\MT@font\@empty
  }
```

`\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
  }
}
```

Consequently, we are the last one to change these commands, therefore there is no need to check whether our definition has survived.[15]

`\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,}}
```

13.4.10 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{%
  \@tempswatrue
  \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
      \@tempswafalse
    \fi
  }%
  \if@tempswa \MT@inlist@true \else \MT@inlist@false \fi
}
```

---

[15] I still don't know whether `microtype` works together with `CJK`

`\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
    \setkeys{MTC}{#1}%
    \selectfont
    \aftergroup\MT@reset@context
  }%
}
```

`\MT@reset@context` We have to reset the font at the end of the group.

```
\def\MT@reset@context{%
  \MT@vinfo{Resetting contexts on line \the\inputlineno
⟨debug⟩  \MessageBreak= \MT@pr@context/\MT@ex@context
⟨debug&beta⟩           /\MT@kn@context/\MT@sp@context
  }%
  \selectfont
}
```

`\MT@setup@contexts` The first time `\microtypecontext` is called, we initialize 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
  }%
  \global\let\MT@check@font\MT@check@font@cx
  \global\let\MT@register@font\MT@register@font@cx
  \global\let\MT@register@subst@font\MT@register@subst@font@cx
  \global\let\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 use the '`@`' instead.

```
      \MT@ifempty{##1}{\def\MT@val{@}}{\KV@@sp@def\MT@val{##1}}%
      \expandafter\ifx\csname MT@\@tempb @context\endcsname\MT@val \else
        \MT@vinfo{--- Changing #1 context to '\MT@val'}%
```

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{2}{>>> added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
        \fi
        \MT@edef@n{MT@\@tempb @context}{\MT@val}%
      \fi
    \fi
  }%
}
```

```
\MT@define@context{protrusion}
\MT@define@context{expansion}
⟨*beta⟩
\MT@define@context{spacing}
\MT@define@context{kerning}
⟨/beta⟩
```

`\MT@pr@context`
`\MT@ex@context`
`\MT@sp@context`
`\MT@kn@context`
`\MT@pr@doc@contexts`
`\MT@ex@doc@contexts`
`\MT@sp@doc@contexts`
`\MT@kn@doc@contexts`
`\MT@extra@context`

Initialize the contexts.

```
\def\MT@pr@context{@}
\def\MT@ex@context{@}
\def\MT@nl@context{@}
\global\def\MT@pr@doc@contexts{{@}}
\global\def\MT@ex@doc@contexts{{@}}
\global\def\MT@nl@doc@contexts{{@}}
⟨*beta⟩
\def\MT@sp@context{@}
\def\MT@kn@context{@}
\global\def\MT@sp@doc@contexts{{@}}
\global\def\MT@kn@doc@contexts{{@}}
⟨/beta⟩
\let\MT@extra@context\@empty
```

13.5 Configuration

13.5.1 Font Sets

`\DeclareMicrotypeSet`
`\DeclareMicrotypeSet*`

Calling this macro will create a comma list for every font characteristic of the form: `\MT`⟨*feature*⟩`list@`⟨*characteristic*⟩`@`⟨*set name*⟩. If the optional argument is empty, lists for both expansion and protrusion will be created.

The third argument must be a list of `key=value` pairs. If a font characteristic 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@declare@sets{pr}{#2}{#3}%
    \MT@declare@sets{ex}{#2}{#3}%
⟨*beta⟩
    \MT@declare@sets{sp}{#2}{#3}%
    \MT@declare@sets{kn}{#2}{#3}%
⟨/beta⟩
  }{%
    \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@is@feature`
`\MT@allowed@features`

Never trust user input: We check whether a feature exists to prevent a rather confusing '`Missing \endcsname inserted`' error message.

```
\def\MT@is@feature#1{%
  \MT@ifdefined@n@TF{MT@rbba@\@tempa}\@firstofone{%
    \MT@error{'\@tempa' is not a defined feature.\MessageBreak
      Ignoring #1}{Defined features are: '\MT@allowed@features'.}%
    \@gobble}%
}
\def\MT@allowed@features{protrusion,expansion%
⟨beta⟩,spacing,kerning%
}
```

`\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 set '\MT@curr@set@name'}%
  }%
  \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 expansion and protrusion 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.

```
    \edef\x{%
      \expandafter\noexpand\expandafter\MT@fix@font@spec
      \expandafter\noexpand\csname MT@#2list@#1@\MT@curr@set@name\endcsname
    }%
    \MT@exp@one@n\MT@addto@setup\x
⟨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@fix@font@spec` Fully expand the font specification and fix catcodes.

```
\def\MT@fix@font@spec#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 Minion-Pro 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 triples of {⟨*lower bound*⟩}{⟨*upper bound*⟩}{⟨*list name*⟩}. For simple sizes, the upper boundary is -1.

```
\def\MT@get@range#1-#2-#3\@nil{%
  \MT@ifempty{#1}{%
    \MT@ifempty{#2}{%
      \let\MT@val\relax
    }{%
      \def\MT@lower{0}%
      \def\MT@val{#2}%
      \MT@get@size
      \edef\MT@upper{\MT@val}%
    }%
  }{%
    \def\MT@val{#1}%
    \MT@get@size
    \ifx\MT@val\relax \else
      \edef\MT@lower{\MT@val}%
      \MT@ifempty{#2}{%
        \MT@ifempty{#3}{%
          \def\MT@upper{-1}%
        }{%
```

2048 is TEX's maximum font size.

```
          \def\MT@upper{2048}%
        }%
      }{%
        \def\MT@val{#2}%
        \MT@get@size
        \ifx\MT@val\relax \else
          \MT@ifgt\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@ifeq\MT@lower\MT@upper{%
            \def\MT@upper{-1}%
          }\relax
        \fi
      }%
    \fi
  }%
}
```

`\MT@get@size` Translate a size selection command and normalize 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,}%
    }%
    \edef\x{%
       \expandafter\noexpand\expandafter\MT@fix@font@spec
       \expandafter\noexpand\csname MT@#1list@font@\MT@curr@set@name\endcsname
    }%
    \MT@exp@one@n\MT@addto@setup\x
⟨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@define@set@keys{pr}
\MT@define@set@keys{ex}
⟨*beta⟩
\MT@define@set@keys{sp}
\MT@define@set@keys{kn}
⟨/beta⟩
```

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@use@set{pr}{#2}%
    \MT@use@set{ex}{#2}%
⟨*beta⟩
    \MT@use@set{sp}{#2}%
    \MT@use@set{kn}{#2}%
⟨/beta⟩
  }{%
    \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@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}{%
    \global\MT@edef@n{MT@#1@setname}{\@tempa}%
    \MT@info{Using \@nameuse{MT@abbr@#1} set '\@tempa'}%
  }{%
    \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
      \global\MT@edef@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}%
  }%
}
```

`\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@set@default@set{pr}{#2}%
    \MT@set@default@set{ex}{#2}%
⟨*beta⟩
    \MT@set@default@set{sp}{#2}%
    \MT@set@default@set{kn}{#2}%
⟨/beta⟩
  }{%
    \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@kn@set`
`\MT@default@sp@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'}%
    \global\MT@edef@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}%
    \global\MT@edef@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'}}%
  \global\MT@edef@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'}%
    \global\let\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
}
```

`\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 neutralize the command.

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

13.5.2 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.

```
⟨*beta⟩
\def\DeclareMicrotypeBabelHook#1#2{%
  \MT@map@clist@n{#1}{%
    \KV@@sp@def\@tempa{##1}%
    \global\MT@def@n{MT@babel@\@tempa}{#2}%
  }%
}
⟨/beta⟩
```

13.5.3 Fine Tuning

The macros `\SetExpansion` and `\SetProtrusion` provide a similar interface for setting the character protrusion resp. expansion factors for a set of fonts.

`\SetProtrusion`
`\MT@pr@c@name`
`\MT@load`
`\MT@extra@factor`
`\MT@extra@preset`
`\MT@extra@unit`
`\MT@extra@context`

This macro accepts three arguments: [options,] set of font characteristics 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).

```
\renewcommand*\SetProtrusion[2][]{%
  \let\MT@pr@c@name\@undefined
  \let\MT@load\@undefined
  \let\MT@extra@factor\@undefined
  \let\MT@extra@unit\@undefined
  \let\MT@extra@preset\@undefined
  \let\MT@extra@context\@empty
```

Parse the optional first argument:

```
  \setkeys{MT@pr@c}{#1}%
```

If the user hasn't specified a name, we will create one.

```
  \MT@get@codes@name{pr}%
  \MT@set@pr@opt
⟨debug⟩\MT@dinfo{1}{creating protrusion list '\MT@pr@c@name'}%
  \def\MT@permutelist{pr@c}%
  \setkeys{MT@pr@c}{#2}%
```

`\MT@permutelist`

We have parsed the second argument, and can now define macros for all permutations of the font characteristics to point to `\MT@pr@c@`⟨*name*⟩, …

```
  \MT@permute
```

… which we can now define to be ⟨*#3*⟩. We want the catcodes to be correct even if this is called in the preamble.

```
  \MT@begin@catcodes
  \MT@set@pr@list
}
```

`\MT@set@pr@list`

Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

```
\def\MT@set@pr@list#1{%
  \global\MT@def@n{MT@pr@c@\MT@pr@c@name}{#1}%
  \MT@end@catcodes
}
```

`\SetExpansion`
`\MT@ex@c@name`
`\MT@load`
`\MT@extra@factor`
`\MT@extra@stretch`
`\MT@extra@shrink`
`\MT@extra@step`
`\MT@extra@auto`
`\MT@extra@preset`
`\MT@extra@context`

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

```
\renewcommand*\SetExpansion[2][]{%
  \let\MT@ex@c@name\@undefined
  \let\MT@load\@undefined
  \let\MT@extra@factor\@undefined
  \let\MT@extra@stretch\@undefined
  \let\MT@extra@shrink\@undefined
  \let\MT@extra@step\@undefined
  \let\MT@extra@auto\@undefined
  \let\MT@extra@preset\@undefined
  \let\MT@extra@context\@empty
  \setkeys{MT@ex@c}{#1}%
  \MT@get@codes@name{ex}%
  \MT@set@ex@opt
⟨debug⟩\MT@dinfo{1}{creating expansion list '\MT@ex@c@name'}%
  \def\MT@permutelist{ex@c}%
  \setkeys{MT@ex@c}{#2}%
  \MT@permute
  \MT@begin@catcodes
  \MT@set@ex@list
}
```

`\MT@set@ex@list` Same story.

```
\def\MT@set@ex@list#1{%
  \global\MT@def@n{MT@ex@c@\MT@ex@c@name}{#1}%
  \MT@end@catcodes
}
```

```
⟨*beta⟩
```

`\SetExtraSpacing`
`\MT@sp@c@name`
`\MT@load`
`\MT@extra@factor`
`\MT@extra@unit`
`\MT@extra@preset`
`\MT@extra@context`

```
\renewcommand*\SetExtraSpacing[2][]{%
  \let\MT@sp@c@name\@undefined
  \let\MT@load\@undefined
  \let\MT@extra@factor\@undefined
  \let\MT@extra@unit\@undefined
  \let\MT@extra@preset\@undefined
  \let\MT@extra@context\@empty
  \setkeys{MT@sp@c}{#1}%
  \MT@get@codes@name{sp}%
  \MT@set@sp@opt
⟨debug⟩\MT@dinfo{1}{creating space list '\MT@sp@c@name'}%
  \def\MT@permutelist{sp@c}%
  \setkeys{MT@sp@c}{#2}%
  \MT@permute
  \MT@begin@catcodes
  \MT@set@sp@list
}
```

`\MT@set@sp@list`

```
\def\MT@set@sp@list#1{%
  \global\MT@def@n{MT@sp@c@\MT@sp@c@name}{#1}%
  \MT@end@catcodes
}
```

`\SetExtraKerning`
`\MT@kn@c@name`
`\MT@load`
`\MT@extra@factor`
`\MT@extra@unit`
`\MT@extra@preset`
`\MT@extra@context`

```
\renewcommand*\SetExtraKerning[2][]{%
  \let\MT@kn@c@name\@undefined
  \let\MT@load\@undefined
  \let\MT@extra@factor\@undefined
  \let\MT@extra@unit\@undefined
  \let\MT@extra@preset\@undefined
  \let\MT@extra@context\@empty
  \setkeys{MT@kn@c}{#1}%
  \MT@get@codes@name{kn}%
  \MT@set@kn@opt
⟨debug⟩\MT@dinfo{1}{creating kerning list '\MT@kn@c@name'}%
  \def\MT@permutelist{kn@c}%
  \setkeys{MT@kn@c}{#2}%
  \MT@permute
  \MT@begin@catcodes
  \MT@set@kn@list
}
```

\MT@set@kn@list

```
\def\MT@set@kn@list#1{%
  \global\MT@def@n{MT@kn@c@\MT@kn@c@name}{#1}%
  \MT@end@catcodes
}
```

```
⟨/beta⟩
```

\MT@get@codes@name Use file name and line number as the list name if the user didn't bother creating one.

```
\def\MT@get@codes@name#1{%
  \MT@ifdefined@n@TF{MT@#1@c@name}{%
    \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
      \MT@warning{Redefining list '\@nameuse{MT@#1@c@name}'}%
    }%
  }{%
    \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
  }%
  \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
```

Now that we know the name, we can cater for any set to be loaded by this list.

```
  \MT@ifdefined@c@T\MT@load{%
    \global\MT@let@nc{MT@#1@c@\MT@curr@set@name load}\MT@load
  }%
}
```

\MT@set@pr@opt Three extra option for protrusion: `factor`, `unit` and `preset`.

```
\def\MT@set@pr@opt{%
  \MT@set@opt@{pr}{factor}%
  \MT@set@opt@{pr}{unit}%
  \MT@set@opt@{pr}{preset}%
}
```

\MT@set@ex@opt The extra options to `\SetExpansion` also have to be dealt with only after we know the name.

```
\def\MT@set@ex@opt{%
  \MT@ifdefined@c@T\MT@extra@factor{%
    \ifnum\MT@extra@factor>\@m
      \MT@warning@nl{Expansion factor \number\MT@extra@factor\space too
        large in list\MessageBreak '\MT@ex@c@name'. Setting it to the
        maximum of 1000}%
      \let\MT@extra@factor\@m
    \fi
    \global\MT@let@nc{MT@ex@c@\MT@ex@c@name @factor}\MT@extra@factor
  }%
  \MT@set@opt@{ex}{stretch}%
  \MT@set@opt@{ex}{shrink}%
  \MT@set@opt@{ex}{step}%
  \MT@set@opt@{ex}{auto}%
  \MT@set@opt@{ex}{preset}%
}
```

```
⟨*beta⟩
```

\MT@set@sp@opt

```
\def\MT@set@sp@opt{%
  \MT@set@opt@{sp}{factor}%
  \MT@set@opt@{sp}{unit}%
  \MT@set@opt@{sp}{preset}%
}
```

`\MT@set@kn@opt`

```
\def\MT@set@kn@opt{%
  \MT@set@opt@{kn}{factor}%
  \MT@set@opt@{kn}{unit}%
  \MT@set@opt@{kn}{preset}%
}
```

```
⟨/beta⟩
```

`\MT@set@opt@`

```
\def\MT@set@opt@#1#2{%
  \MT@ifdefined@n@T{MT@extra@#2}{%
    \global\MT@let@nn{MT@#1@c@\csname MT@#1@c@name\endcsname @#2}{MT@extra@#2}%
  }%
}
```

`\MT@define@code@key` Define the keys for expansion and protrusion character code lists.

```
\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\romannumeral\@tempcnta}{\MT@val}%
      \advance\@tempcnta \@ne
    }%
  }%
}
```

`\MT@define@code@key@size`

```
\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}{\csname MT@#1@name\endcsname}}}%
      \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
      \global\MT@edef@n{MT@\MT@permutelist @\@tempb}%
        {\csname MT@\MT@permutelist @name\endcsname}%
⟨*debug⟩
        \MT@dinfo@nl{1}{initializing: 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 @sizes\endcsname
        {{{\MT@val}{\m@ne}{\csname MT@#1@name\endcsname}}}%
    }%
  }%
}
```

`\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@declare@codes`

```
\def\MT@declare@codes#1{%
  \define@key{MT@#1@c}{name}[]{%
    \MT@ifempty{##1}\relax{%
      \MT@def@n{MT@#1@c@name}{##1}%
    }%
  }%
  \define@key{MT@#1@c}{load}[]{%
    \MT@ifempty{##1}\relax{%
      \def\MT@load{##1}%
    }%
  }%
  \define@key{MT@#1@c}{factor}[]{%
    \MT@ifempty{##1}\relax{%
      \def\MT@extra@factor{##1 }%
    }%
  }%
  \MT@define@code@key{encoding}{#1@c}%
  \MT@define@code@key{family}{#1@c}%
  \MT@define@code@key{series}{#1@c}%
  \MT@define@code@key{shape}{#1@c}%
  \MT@define@code@key@size{#1@c}%
  \MT@define@code@key@font{#1@c}%
  \define@key{MT@#1@c}{preset}[]{%
    \MT@ifempty{##1}\relax{%
      \def\MT@extra@preset{##1}%
    }%
  }%
```

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@declare@codes{pr}
\MT@declare@codes{ex}
⟨*beta⟩
\MT@declare@codes{sp}
\MT@declare@codes{kn}
⟨/beta⟩
```

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

```
\define@key{MT@pr@c}{unit}[character]{%
  \let\MT@extra@unit\@empty
  \KV@@sp@def\@tempa{#1}%
  \MT@ifstreq\@tempa{relative}{%
    \MT@warning{Value 'relative' for key 'unit' is deprecated.\MessageBreak
      Use 'unit=character' instead. For now, I'll do it\MessageBreak
      for you}%
    \def\@tempa{character}%
  }\relax
  \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{%
      \let\MT@extra@unit\@tempa
    }{%
      \MT@warning{'\@tempa' is not a dimension.\MessageBreak
        Ignoring it and setting values relative to\MessageBreak
        character widths}%
    }%
  }%
}
```

`\MT@define@key@unit` Spacing and kerning codes may additionally be relative to space dimensions.

```
⟨*beta⟩
\def\MT@define@key@unit#1{%
  \define@key{MT@#1@c}{unit}[space]{%
    \let\MT@extra@unit\@empty
    \KV@@sp@def\@tempa{##1}%
    \MT@ifstreq\@tempa{relative}{%
      \MT@warning{Value 'relative' for key 'unit' is deprecated.\MessageBreak
        Use 'unit=character' instead. For now, I'll do it\MessageBreak
        for you}%
      \def\@tempa{character}%
    }\relax
    \MT@ifstreq\@tempa{character}\relax{%
      \let\MT@extra@unit\m@ne
      \MT@ifstreq\@tempa{space}\relax{%
        \MT@ifdimen\@tempa{%
          \let\MT@extra@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}
⟨/beta⟩
```

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

```
\def\MT@define@ex@c@key#1{%
  \define@key{MT@ex@c}{#1}[]{%
    \MT@ifempty{##1}\relax{%
      \MT@ifnumber{##1}{%
```

A space terminates the number.

```
        \MT@def@n{MT@extra@#1}{##1 }%
      }{%
        \MT@warning{%
          Value '##1' for option '#1' is not a number.\MessageBreak
          Ignoring it}%
      }%
    }%
  }%
}
```

```
\MT@define@ex@c@key{stretch}
\MT@define@ex@c@key{shrink}
\MT@define@ex@c@key{step}
\define@key{MT@ex@c}{auto}[true]{%
  \KV@@sp@def\@tempa{#1}%
  \csname if\@tempa\endcsname
```

Don't alter `\MT@extra@auto` for pdfTeX version older than 1.20.

```
    \MT@requires@pdftex4{%
      \def\MT@extra@auto{autoexpand}%
    }{%
      \MT@warning{pdfTeX too old for automatic font expansion}%
    }
  \else
    \MT@requires@pdftex4{%
      \let\MT@extra@auto\@empty
    }\relax
  \fi
}
```

13.5.4 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.

```
\renewcommand*\DeclareCharacterInheritance[1][]{%
  \KV@@sp@def\@tempa{#1}%
  \MT@begin@catcodes
  \MT@set@inh@list
}
```

`\MT@set@inh@list` Safe category codes.

```
\def\MT@set@inh@list#1#2{%
  \MT@ifempty\@tempa{%
    \MT@declare@char@inh{pr}{#1}{#2}%
    \MT@declare@char@inh{ex}{#1}{#2}%
⟨*beta⟩
    \MT@declare@char@inh{sp}{#1}{#2}%
    \MT@declare@char@inh{kn}{#1}{#2}%
⟨/beta⟩
  }{%
    \MT@map@clist@c\@tempa{{%
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \MT@is@feature{inheritance declaration for '#1'}{%
          \MT@exp@one@n\MT@declare@char@inh
            {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
        }%
      }%
    }}%
  }%
  \MT@end@catcodes
}
```

`\MT@declare@char@inh` The optional argument may be used to restrict the inheritance list to protrusion or expansion.

```
\def\MT@declare@char@inh#1#2#3{%
  \MT@let@nc{MT@#1@inh@name}\@undefined
  \MT@get@inh@name{#1}%
⟨debug⟩\MT@dinfo{1}{creating inheritance list '\@nameuse{MT@#1@inh@name}'}%
  \global\MT@def@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
  \def\MT@permutelist{#1@inh}%
  \setkeys{MT@#1@inh}{#2}%
  \MT@permute
}
```

`\MT@get@inh@name` The inheritance lists cannot be named by the user.

```
\def\MT@get@inh@name#1{%
  \MT@edef@n{MT@#1@inh@name}%
    {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
}
```

`\MT@define@inh@key@encoding` Parse the first argument. `\DeclareCharacterInheritance` may also be set up for various combinations.

```
\def\MT@define@inh@key@encoding#1{%
  \define@key{MT@#1}{encoding}[]{%
    \def\MT@val{##1}%
    \expandafter\MT@encoding@check\MT@val,\@nil
    \MT@get@highlevel{encoding}%
    \MT@edef@n{MT@tempencoding\romannumeral1}{\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}%
  }%
}
```

\MT@define@inh@keys

```
\def\MT@define@inh@keys#1{%
  \MT@define@inh@key@encoding{#1@inh}%
```

For the rest, we can reuse the key setup from `\SetProtrusion` resp. `\SetExpansion`.

```
  \MT@define@code@key{family}{#1@inh}%
  \MT@define@code@key{series}{#1@inh}%
  \MT@define@code@key{shape}{#1@inh}%
  \MT@define@code@key@size{#1@inh}%
  \MT@define@code@key@font{#1@inh}%
}
\MT@define@inh@keys{pr}
\MT@define@inh@keys{ex}
⟨*beta⟩
\MT@define@inh@keys{sp}
\MT@define@inh@keys{kn}
⟨/beta⟩
```

\MT@inh@do Parse the second 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@pr@codes` resp. `\MT@set@ex@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@inh@name @\MT@val @\endcsname
              {{\MT@char}}%
          \fi
        \fi
      }%
⟨*debug⟩
      \MT@dinfo@nl{2}{children of #1 (\MT@val):
                      \@nameuse{MT@inh@\MT@inh@name @\MT@val @}}%
⟨/debug⟩
    \fi
  \fi
}
```

13.5.5 Permutation

`\MT@permute` `\MT@permute@` `\MT@permute@@` `\MT@permute@@@` `\MT@permute@@@@`

Calling `\MT@permute` will define commands for all permutations of the specified font characteristics 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@permute@reset
}
\def\MT@permute@{%
  \let\MT@cnt@family\@ne
  \MT@permute@@
  \MT@increment\MT@cnt@encoding
  \MT@ifdefined@n@T{MT@tempencoding\romannumeral\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\romannumeral\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\romannumeral\MT@cnt@series}%
    \MT@permute@@@
}
\def\MT@permute@@@@{%
  \MT@permute@@@@@
  \MT@increment\MT@cnt@shape
  \MT@ifdefined@n@T{MT@tempshape\romannumeral\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}{initializing: use list for font \@tempa,\MessageBreak
                   sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
                                  @sizes\endcsname}%
⟨/debug⟩
      }{%
```

Only one list should 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}{initializing: use list for font \@tempa
                        \ifx\MT@extra@context\@empty\else\MessageBreak
                          (context: \MT@extra@context)\fi}%
⟨/debug⟩
      }%
      \global\MT@edef@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\romannumeral\@tempcnta}%
    {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\romannumeral\@tempcnta\endcsname}}%
    {\MT@let@nc{MT@temp#1}\@empty}%
}
```

`\MT@permute@reset` Reset the commands.

```
\def\MT@permute@reset{%
  \MT@permute@reset@{encoding}%
  \MT@permute@reset@{family}%
  \MT@permute@reset@{series}%
  \MT@permute@reset@{shape}%
  \let\MT@tempsize\@undefined
}
```

`\MT@permute@reset@`

```
\def\MT@permute@reset@#1{%
  \@tempcnta=\@ne
  \MT@loop
    \MT@let@nc{MT@temp#1\romannumeral\@tempcnta}\@undefined
    \advance\@tempcnta\@ne
    \MT@ifdefined@n@TF{MT@temp#1\romannumeral\@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}%
  \@tempswafalse
  \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@ifeq{#2}\m@ne{%
    \MT@ifeq\@tempc\m@ne{%
```

- Both items are simple sizes.

```
          \MT@ifeq\@tempb{#1}\@tempswatrue\relax
        }{%
```

- Item in list is a simple size, new item is a range.

```
          \MT@ifgt\@tempb{#1}\relax{%
            \MT@ifgt\@tempc{#1}{%
              \@tempswatrue
              \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
            }\relax
          }%
        }%
      }{%
        \MT@ifeq\@tempc\m@ne{%
```

- Item in list is a range, new item is a simple size.

```
          \MT@iflt\@tempb{#2}{%
            \MT@iflt\@tempb{#1}\relax\@tempswatrue
          }\relax
        }{%
```

- Both items are ranges.

```
          \MT@iflt\@tempb{#2}{%
            \MT@ifgt\@tempc{#1}{%
              \@tempswatrue
              \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
            }\relax
          }\relax
        }%
      }%
      \if@tempswa
        \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
    }
```

13.6 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` and `activate`, and `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 recognized.\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
⟨*beta⟩
```

`\MT@define@optionX@` The same for `spacing` and `kerning`, which do not have a `nocompatibility` level.

```
\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 recognized.\MessageBreak
              Use either 'true' or 'false'}%
          }%
        }%
        \ifnum\@tempcnta>\m@ne
          #2\relax
        \fi
      }%
    }%
  }%
}
```

```
\MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
\MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
                             \pdfappendkern \@tempcnta}
⟨/beta⟩
\define@key{MTX}{activate}[]{%
  \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.)*

```
\def\MT@gobblethree#1#2#3{}
\let\MT@saved@setupfont\MT@setupfont
\define@key{MTX}{disable}[]{%
  \MT@info{Inactivate '\MT@MT' package}%
  \let\MT@setupfont\MT@gobblethree
}
\define@key{MTX}{enable}[]{%
  \MT@info{Reactivate '\MT@MT' package}%
  \let\MT@setupfont\MT@saved@setupfont
}
```

13.7 Package Options

13.7.1 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}{%
                  \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
                }{%
                  \global\MT@edef@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` (and `spacing`?).

```
\define@key{MT}{activate}[]{%
  \setkeys{MT}{protrusion={#1}}%
  \setkeys{MT}{expansion={#1}}%
}
⟨*beta⟩
```

`\MT@define@option@` `spacing` and `kerning` 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}{%
              \global\MT@edef@n{MT@\@tempb @setname}{\MT@val}%
            }{%
              \global\MT@edef@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}
⟨/beta⟩
```

`\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}%
⟨beta⟩{babel}%
}\MT@def@simple@bool@opt
```

The `DVIoutput` option will change `\pdfoutput` immediately to minimize 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 know whether it is actually 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@microtype
      \let\MT@setup@microtype\@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`, `letterspacing`.

```
\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@ifnumber\@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}}%
    }%
  }%
}
```

```
\MT@map@tlist@n{{stretch}{shrink}{step}%
⟨beta⟩{letterspacing}%
}\MT@def@num@opt
```

`factor` will define the protrusion factor only.

```
\define@key{MT}{factor}[]{%
  \MT@ifempty{#1}%
    {\let\@tempa\MT@factor@default}%
    {\def\@tempa{#1 }}%
  \MT@ifnumber\@tempa{%
    \MT@edef@n{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 codes.

```
\define@key{MT}{unit}[]{%
  \MT@ifempty{#1}%
    {\def\@tempa{character}}%
    {\KV@@sp@def\@tempa{#1}}%
  \MT@ifstreq\@tempa{relative}{%
    \MT@warning{Value 'relative' for option 'unit' is deprecated.\MessageBreak
      Use 'unit=character' instead. For now, I'll do it\MessageBreak
      for you}%
    \def\@tempa{character}%
  }\relax
  \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}%
    }%
  }%
}
```

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=\z@ \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@ifdefined@c@TF\MT@default@pr@set\relax{\gdef\MT@default@pr@set{all}}
\MT@ifdefined@c@TF\MT@default@ex@set\relax{\gdef\MT@default@ex@set{all}}
⟨*beta⟩
\MT@ifdefined@c@TF\MT@default@sp@set\relax{\gdef\MT@default@sp@set{all}}
\MT@ifdefined@c@TF\MT@default@kn@set\relax{\gdef\MT@default@kn@set{all}}
⟨/beta⟩
```

13.7.2 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 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}}
\@ifundefined{Microtype@Hook}
  {\let\Microtype@Hook\MinionPro@MT@Hook}
  {\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}}
```

`\MicroType@Hook` with a capital `T` 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@nl{%
    Command \string\MicroType@Hook\space is deprecated.\MessageBreak
    Use \string\Microtype@Hook\space instead}\MicroType@Hook}
\MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

13.7.3 Processing the Options

`\MT@ProcessOptionsWithKV`

Parse options.

```
\def\MT@ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\KVo@tempa\@empty
  \MT@map@clist@c\@classoptionslist{%
    \def\CurrentOption{##1}%
    \MT@ifdefined@n@T{KV@#1@\CurrentOption}{%
      \edef\KVo@tempa{\KVo@tempa,\CurrentOption,}%
      \@expandtwoargs\@removeelement\CurrentOption
        \@unusedoptionlist\@unusedoptionlist
    }%
  }%
  \edef\KVo@tempa{%
    \noexpand\setkeys{#1}{%
      \KVo@tempa\@ptionlist{\@currname.\@currext}%
    }%
  }%
  \KVo@tempa
  \AtEndOfPackage{\let\@unprocessedoptions\relax}%
  \let\CurrentOption\@empty
}
```

```
\MT@ProcessOptionsWithKV{MT}
```

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 3171 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
⟨*beta⟩
  \MT@spacingfalse
  \MT@kerningfalse
  \MT@babelfalse
⟨/beta⟩
  \let\MT@setupfont\relax
  \def\DeclareMicrotypeSet#1#{\@gobbletwo}
  \renewcommand*\UseMicrotypeSet[2][]{}
  \renewcommand*\SetProtrusion[3][]{}
  \renewcommand*\SetExpansion[3][]{}
⟨*beta⟩
  \renewcommand*\SetExtraSpacing[3][]{}
  \renewcommand*\SetExtraKerning[3][]{}
⟨/beta⟩
  \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=\z@
    \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=\z@ DVI \else PDF \fi output%
              \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
```

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}
```

We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by `\DeclareMicrotypeSetDefault`.

```
  \MT@ifdefined@c@TF\MT@pr@setname{%
    \MT@info@nl{Using protrusion set '\MT@pr@setname'}%
  }{%
    \global\let\MT@pr@setname\MT@default@pr@set
    \MT@info@nl{Using default protrusion set '\MT@pr@setname'}%
  }
\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=\z@
        \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.

```
    }{%
      \ifMT@opt@auto
        \MT@warning@nl{%
          The pdftex you are using is too old for automatic\MessageBreak
          font expansion. I will switch it off and hope that\MessageBreak
          expanded fonts are available on your system.\MessageBreak
          Install pdftex version 1.20 or newer}
      \fi
      \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@ifdefined@c@TF\MT@ex@setname{%
      \MT@info@nl{Using expansion set '\MT@ex@setname'}%
    }{%
      \global\let\MT@ex@setname\MT@default@ex@set
      \MT@info@nl{Using default expansion set '\MT@ex@setname'}%
    }
```

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
\fi
}
⟨*beta⟩
\MT@requires@pdftex6{%
  \MT@addto@setup{%
```

Spacing.

```
    \ifMT@spacing
      \edef\MT@active@features{\MT@active@features,sp}
      \pdfadjustinterwordglue\@ne
      \MT@info@nl{Adjustment of interword spacing enabled}
      \MT@ifdefined@c@TF\MT@sp@setname{%
        \MT@info@nl{Using spacing set '\MT@sp@setname'}%
      }{%
        \global\let\MT@sp@setname\MT@default@sp@set
        \MT@info@nl{Using default spacing set '\MT@sp@setname'}%
      }
```

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 this message from Philipp Lehman: <`ddtbaj$rob$1@online.de`> on August 16, 2005.

```
      \ifnum\sfcode'\. > 1500
        \ifMT@babel \else
          \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 '\MT@MT context{spacing=nonfrench}'\MessageBreak
              to your preamble}%
          }%
        \fi
      \fi
    \else
      \let\MT@spacing\relax
      \MT@info@nl{No adjustment of interword spacing}
    \fi
```

Kerning is always active (because of letterspacing). Hence, we also don't set `\MT@kerning` to `\relax`.

```
    \edef\MT@active@features{\MT@active@features,kn}
    \ifMT@kerning
      \pdfprependkern\@ne
      \pdfappendkern\@ne
      \MT@info@nl{Adjustment of character kerning enabled}
      \MT@ifdefined@c@TF\MT@kn@setname{%
        \MT@info@nl{Using kerning set '\MT@kn@setname'}%
      }{%
        \global\let\MT@kn@setname\MT@default@kn@set
        \MT@info@nl{Using default kerning set '\MT@kn@setname'}%
      }
    \else
      \MT@info@nl{No adjustment of character kerning}
    \fi
    \ifnum\MT@letterspacing=\m@ne
      \let\MT@letterspacing\MT@letterspacing@default
    \fi
  }
```

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

```
}{%
  \MT@addto@setup{%
    \ifMT@spacing
      \MT@warning@nl{Adjustment of interword spacing only works with\MessageBreak
        pdftex version 1.4x or newer. Switching it off}%
    \else
      \MT@info@nl{No adjustment of interword spacing}
    \fi
    \MT@spacingfalse
    \let\MT@spacing\relax
    \ifMT@kerning
      \MT@warning@nl{Character kerning only works with\MessageBreak
        pdftex version 1.4x or newer. Switching it off}%
    \else
      \MT@info@nl{No adjustment of character kerning}
    \fi
    \MT@kerningfalse
    \let\MT@kerning\relax
  }
}
```

Interaction with `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@ifdefined@n@TF{MT@babel@#1}{%
          \MT@vinfo{Changing to language '#1' on line \the\inputlineno}%
          \expandafter\MT@exp@one@n\expandafter\microtypecontext
              \csname MT@babel@#1\endcsname
        }{%
          \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
        }%
      }
      \let\MT@orig@foreign@language\foreign@language
      \def\foreign@language#1{%
        \MT@orig@foreign@language{#1}%
        \MT@ifdefined@n@TF{MT@babel@#1}{%
          \MT@vinfo{Changing to context '#1' on line \the\inputlineno}%
          \expandafter\MT@exp@one@n\expandafter\microtypecontext
              \csname MT@babel@#1\endcsname
        }{%
          \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
        }%
      }
```

Disable `babel`’s active characters.

```
      \ifMT@kerning
        \@tempswafalse
        \@ifpackagewith{babel}{french}\@tempswatrue\relax
        \@ifpackagewith{babel}{frenchb}\@tempswatrue\relax
        \@ifpackagewith{babel}{francais}\@tempswatrue\relax
        \if@tempswa
          \NoAutoSpaceBeforeFDP
          \MT@warning@nl{Switching off French babel's active punctuation characters}%
        \fi
      \fi
    }{%
      \MT@warning@nl{You did not load the babel package.\MessageBreak
        The 'babel' option won't have any effect}
    }
  \fi
}
⟨/beta⟩
```

Finally, 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
}
```

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
```

This is the current file.

```
\edef\MT@curr@file{\jobname.tex}
⟨/package⟩
```

That was that.

14 Configuration Files

Let’s now write the font configuration files.

```
⟨*config⟩

```

14.1 Font Sets

We first declare some sets in the main configuration file.

```
⟨*m-t⟩
%%% ----------------------------------------------------------------------
%%% FONT SETS

\DeclareMicrotypeSet{all}
   { }

\DeclareMicrotypeSet{allmath}
   { encoding = {T1,LY1,OT1,OT4,T5,TS1,OML,OMS,U} }

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

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

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

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

```

The default sets.

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

\DeclareMicrotypeSetDefault[protrusion]{alltext}
\DeclareMicrotypeSetDefault[expansion] {basictext}
⟨*beta⟩
\DeclareMicrotypeSetDefault[spacing]   {basictext}
\DeclareMicrotypeSetDefault[kerning]   {alltext}
⟨/beta⟩

```

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. The packages `pxfonts` and `txfonts` fonts inherit Palatino and Times settings respectively, also the `qfonts` package which provides both, and `mathdesign`'s Charter should be the same as Bitstream Charter.

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

\DeclareMicrotypeAlias{lmr} {cmr}  % lmodern
\DeclareMicrotypeAlias{aer} {cmr}  % ae
\DeclareMicrotypeAlias{zer} {cmr}  % zefonts
\DeclareMicrotypeAlias{cmor}{cmr}  % eco
\DeclareMicrotypeAlias{hfor}{cmr}  % hfoldsty
\DeclareMicrotypeAlias{pxr} {ppl}  % pxfonts
\DeclareMicrotypeAlias{qpl} {ppl}  % qfonts/QuasiPalatino
\DeclareMicrotypeAlias{txr} {ptm}  % txfonts
\DeclareMicrotypeAlias{qtm} {ptm}  % qfonts/QuasiTimes
\DeclareMicrotypeAlias{mdbch}{bch} % MathDesign/Charter

```

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

```
⟨*beta⟩
```

14.2 Interaction with `babel`

```
%%% ----------------------------------------------------------------------
%%% INTERACTION WITH THE 'babel' PACKAGE

\DeclareMicrotypeBabelHook
   {french,francais}
   {kerning=french, spacing=}

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

\DeclareMicrotypeBabelHook
   {turkish}
   {kerning=turkish, spacing=}

⟨/beta⟩
```

14.3 Note on Admissible Characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```
\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#
```

Comma and equal sign must be guarded with braces (‘`{,}`’, ‘`{=}`’) to keep `keyval` happy.

Character commands are allowed as far as they have been defined in the proper LaTeX way, that is, when they have been assigned a slot in the font encoding with `\DeclareTextSymbol` or `\DeclareTextComposite`. Characters defined via `\chardef` are also possible.

Ligatures and `\mathchardef`ed symbols have to be specified numerically. Of course, numerical identification is possible in any other case, too.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed.

14.4 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.

```
%%% ----------------------------------------------------------------------
%%% CHARACTER INHERITANCE

```

14.4.1 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},
   }

```

14.4.2 T1

Candidates here: `028` (‘fi’), `029` (‘fl’), `030` (‘ffi’), `031` (‘ffl’), `156` (‘IJ’ ligature), `188` (‘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},
   }

```

14.4.3 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}},
  }

```

14.4.4 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},
  }

```

14.4.5 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}},
  }

```

14.5 Font Expansion

These are Hàn Thế Thành’s original expansion settings. They are used for all fonts (until somebody shows mercy and creates font-specific settings).

```
%%% ----------------------------------------------------------------------
%%% EXPANSION SETTINGS

\SetExpansion
⟨m-t⟩   [ name     = default      ]
⟨bch⟩   [ name     = bch-default  ]
⟨cmr⟩   [ name     = cmr-default  ]
⟨pad⟩   [ name     = pad-default  ]
⟨pmn⟩   [ name     = pmn-default  ]
⟨ppl⟩   [ name     = ppl-default  ]
⟨ptm⟩   [ name     = ptm-default  ]
⟨m-t⟩   { encoding = {OT1,OT4,T1,LY1} }
⟨!m-t⟩   { encoding = {OT1,T1,LY1},
⟨bch⟩     family   = bch }
⟨cmr⟩     family   = cmr }
⟨pad⟩     family   = {pad,padx,padj} }
⟨pmn⟩     family   = {pmn,pmnx,pmnj} }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
   {
     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⟩
```

14.6 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},
   }
```

We also create configuration files for the fonts Bitstream Charter (NFSS code `bch`), Computer Modern Roman (`cmr`), Palatino (`ppl`, `pplx`, `pplj`), Times (`ptm`, `ptmx`, `ptmj`), Adobe Garamond (`pad`, `padx`, `padj`) and Minion[16] (`pmnx`, `pmnj`), and for the AMS math fonts (`msa`, `msb`, `euf`, `eus`).

14.6.1 Default

The default settings always use the most moderate value.

```
⟨*!cfg-u⟩
\SetProtrusion
⟨m-t⟩   [ name     = default ]
⟨bch⟩   [ name     = bch-default ]
⟨cmr⟩   [ name     = cmr-default ]
⟨pad⟩   [ name     = pad-default ]
⟨pmn⟩   [ name     = pmnj-default ]
⟨ppl⟩   [ name     = ppl-default ]
⟨ptm⟩   [ name     = ptm-default ]
⟨m-t⟩   { encoding = OT1     }
⟨cmr⟩   { }
⟨bch|pad|pmn⟩   { 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} }
   {
     A = {50,50},
⟨m-t|pad|ptm⟩   \AE = {50,  },
⟨bch|pad|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,  },
     K = {  ,50},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     L = {  ,50},
```

---

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

```
⟨ptm⟩      L = {   ,80},
⟨bch|pad|pmn⟩      O = {50,50},
⟨pad|pmn⟩    \OE = {50,  },
⟨bch|pad|pmn⟩      Q = {50,70},
⟨bch⟩      R = {   ,50},
     T = {50,50},
     V = {50,50},
     W = {50,50},
     X = {50,50},
⟨m-t|bch|cmr|pad|pmn|ppl⟩      Y = {50,50},
⟨ptm⟩      Y = {80,80},
     k = {   ,50},
⟨pmn⟩      l = {   ,-50},
⟨pad|ppl⟩      p = {50,50},
⟨pad|ppl⟩      q = {50,  },
     r = {   ,50},
⟨cmr|pad|pmn⟩      t = {   ,70},
⟨bch⟩      t = {   ,50},
     v = {50,50},
     w = {50,50},
     x = {50,50},
⟨m-t|bch|pad|pmn⟩      y = {   ,50},
⟨cmr|ppl|ptm⟩      y = {50,70},

⟨cmr⟩      0 = {   ,50},
⟨m-t⟩      1 = {50,50},
⟨bch|pad|ptm⟩      1 = {150,150},
⟨cmr⟩      1 = {100,200},
⟨pmn⟩      1 = {   ,50},
⟨ppl⟩      1 = {100,100},
⟨bch|cmr|pad⟩      2 = {50,50},
⟨cmr|pad⟩      3 = {50,50},
⟨bch|pmn⟩      3 = {50,  },
⟨m-t|pad⟩      4 = {50,50},
⟨bch⟩      4 = {100,50},
⟨cmr⟩      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⟩      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⟩      . = {  ,700},
⟨bch⟩      . = {  ,600},
    {,}= {  ,500},
⟨m-t|cmr|pad|pmn|ppl|ptm⟩      : = {  ,500},
⟨bch⟩      : = {  ,400},
⟨m-t|bch|pad|pmn|ptm⟩      ; = {  ,300},
⟨cmr|ppl⟩      ; = {  ,500},
     ! = {  ,100},
⟨m-t|pad|pmn|ptm⟩      ? = {  ,100},
⟨bch|cmr|ppl⟩      ? = {  ,200},
⟨pmn⟩      " = {300,300},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     @ = {50,50},
⟨ptm⟩      @ = {100,100},
      ~ = {200,250},
⟨pad|ppl|ptm⟩     & = {50,100},
⟨m-t|cmr|pad|pmn⟩     \% = {50,50},
⟨bch⟩     \% = {   ,50},
⟨ppl|ptm⟩     \% = {100,100},
⟨m-t|ppl|ptm⟩     * = {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},
⟨m-t|pad|pmn|ptm⟩     ( = {100,   },    ) = {   ,200},
⟨bch⟩      ( = {200,   },    ) = {   ,200},
⟨cmr|ppl⟩     ( = {100,   },    ) = {   ,300},
⟨bch|pmn⟩     [ = {100,   },    ] = {   ,100},

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

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},
⟨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},
   }

```

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
%    "04 = {   ,   }, % \Xi
%    "05 = {   ,   }, % \Pi
     "06 = { 50, 50}, % \Sigma
     "07 = {100,100}, % \Upsilon
     "08 = { 50, 50}, % \Phi
     "09 = { 50, 50}, % \Psi
%    "0A = {   ,   }, % \Omega
   }

⟨/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,
⟨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 ]
⟨m-t⟩   { encoding = {T1,LY1}    }
⟨bch|cmr|pad|pmn|ppl⟩   { encoding = {T1,LY1},
⟨ptm⟩   { 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} }
   {
⟨cmr⟩     \AE = {50,  },
⟨bch⟩     \OE = {50,  },
⟨pmn⟩     \TH = {  ,50},
⟨m-t|bch|pad|pmn|ppl|ptm⟩       _ = {100,100},
⟨cmr⟩       _ = {200,200},
⟨m-t|pad|pmn|ptm⟩     \textbackslash    = {100,200},
⟨bch⟩     \textbackslash    = {150,200},
⟨cmr|ppl⟩     \textbackslash    = {200,300},
⟨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⟩     \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},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch|pmn⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨pad⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ppl|ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨m-t|bch|cmr|pad|pmn|ppl⟩     \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨m-t|cmr|pad|ppl|ptm⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨bch|pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t|bch|cmr|pad|ppl|ptm⟩     \textless         = {200,100},   \textgreater       = {100,200},
⟨pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100},
⟨pmn⟩     \textvisiblespace = {100,100}, % not in LY1
%   \dh = {  ,  },
%   \th = {  ,  },
%   \NG = {  ,  },
%   \ng = {  ,  },
%   \textasciicircum = {  ,  },
%   \textbar = {  ,  },
%   \textsterling = {  ,  }, % also in TS1
%   \textsection = {  ,  }, % also in TS1
  }

⟨*cmr|bch⟩
```

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.

```
\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⟩
```

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⟩
⟨*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             = {100,100},
     \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},
%    \textbrokenbar            = {   ,   },
%    \textyen                  = {   ,   },
%    \textfractionsolidus      = {   ,   },
%    \textordfeminine          = {   ,   },
%    \textordmasculine         = {   ,   },
%    \textmu                   = {   ,   },
%    \textparagraph            = {   ,   },
%    \textonequarter          = {   ,   },
%    \textonehalf             = {   ,   },
%    \textthreequarters       = {   ,   },
   }

⟨/ptm⟩
```

14.6.2 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   ]
⟨m-t|bch|pad|pmn⟩   { encoding = OT1,
⟨cmr⟩   { }
⟨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},
⟨!cmr⟩     shape    = {it,sl}  }
   {
⟨cmr|ptm⟩     A = {100,50},
⟨pad|pmn⟩     A = {50,  },
⟨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⟩     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⟩     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},
⟨cmr|ptm⟩      M = {50,  },
⟨pmn⟩      M = {  ,-30},
⟨cmr|ptm⟩      N = {50,  },
⟨pmn⟩      N = {  ,-30},
⟨bch|pmn|ppl|ptm⟩      O = {50,  },
⟨cmr|pad⟩      O = {100, },
⟨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, },
⟨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⟩      T = {70,  },
⟨cmr|pad|ppl|ptm⟩      T = {100, },
⟨cmr|pad|ppl|ptm⟩      U = {50,  },
⟨pmn⟩      U = {50,-50},
⟨cmr|pad|pmn⟩      V = {100, },
⟨ppl|ptm⟩      V = {100,50},
⟨cmr|pad|pmn⟩      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⟩      v = {50,  },
⟨bch⟩      w = {  ,50},
⟨pmn⟩      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, },
⟨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⟩      . = { ,700},
⟨m-t|cmr|pad|pmn|ppl⟩     {,}= { ,500},
⟨bch⟩     {,}= { ,600},
⟨ptm⟩     {,}= { ,700},
⟨m-t|cmr|pad|ppl⟩      : = { ,300},
⟨bch⟩      : = { ,400},
⟨pmn⟩      : = { ,200},
⟨ptm⟩      : = { ,500},
⟨m-t|cmr|pad|ppl⟩      ; = { ,300},
⟨bch⟩      ; = { ,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},
⟨m-t|cmr|pad|pmn⟩     \% = {100, },
⟨bch⟩     \% = {50,50},
⟨ppl|ptm⟩     \% = {100,100},
⟨m-t|pmn|ppl⟩      * = {200,200},
⟨bch⟩      * = {300,200},
⟨cmr⟩      * = {400,100},
⟨pad⟩      * = {500,100},
⟨ptm⟩      * = {400,200},
⟨m-t|cmr|pmn|ppl⟩      + = {150,200},
⟨bch⟩      + = {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⟩      ~ = {150,150},
⟨cmr|pad|pmn|ppl|ptm⟩      ~ = {200,150},
      ( = {200, },    ) = {  ,200},
⟨m-t|cmr|pad|ppl|ptm⟩      / = {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},
⟨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⟩     \textendash       = {300,300},   \textemdash        = {200,200},
⟨m-t|bch|pmn⟩     \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},
   }

⟨*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,
⟨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   ]
⟨m-t|bch|cmr|pad|pmn|ppl⟩   { encoding = {T1,LY1},
⟨ptm⟩   { 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},
⟨!cmr⟩     shape    = {it,sl}  }
⟨cmr⟩     shape    = it       }
   {
⟨m-t|bch|pmn⟩     _ = {  ,100},
⟨cmr⟩     _ = {100,200},
⟨pad|ppl|ptm⟩     _ = {100,100},
⟨cmr⟩     \AE = {100,   },
⟨bch⟩     \OE = { 50,   },
⟨cmr⟩     \OE = {100,   },
⟨m-t|pad|ppl|ptm⟩     \textbackslash    = {100,200},
⟨cmr⟩     \textbackslash    = {300,300},
⟨bch⟩     \textbackslash    = {150,150},
⟨pmn⟩     \textbackslash    = {100,150},
⟨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},
⟨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},
⟨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},
⟨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},
⟨m-t|pad|ppl⟩     \textexclamdown   = {100,   },   \textquestiondown  = {200,   },
⟨cmr|ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨pmn⟩     \textexclamdown   = {-50,   },   \textquestiondown  = {-50,   },
⟨m-t|ppl⟩     \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},
  }

⟨*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             = {100,150},
    \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⟩
```

14.6.3 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).

```
\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},
   }

⟨*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},
   }

```

14.6.4 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⟩
```

14.6.5 `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 ]
```

```
⟨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} }
    {
⟨cmr⟩     \textquotestraightbase    = {300,300},
⟨pad|pmn⟩     \textquotestraightbase    = {400,400},
⟨cmr|pmn⟩     \textquotestraightdblbase = {300,300},
⟨pad⟩     \textquotestraightdblbase = {400,400},
⟨bch|cmr|pad|pmn⟩     \texttwelveudash          = {200,200},
⟨bch|cmr|pad|pmn⟩     \textthreequartersemdash  = {150,150},
⟨cmr|pmn⟩     \textquotesingle          = {300,400},
⟨pad⟩     \textquotesingle          = {400,500},
⟨ptm⟩     \textquotesingle          = {500,500},
⟨bch|cmr|pmn⟩     \textasteriskcentered     = {200,300},
⟨pad⟩     \textasteriskcentered     = {300,300},
⟨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},
⟨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⟩     \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},
⟨bch⟩     \textbullet               = {200,200},
⟨cmr|pad|pmn⟩     \textbullet               = {   ,100},
⟨ptm⟩     \textbullet               = {150,150},
⟨bch|cmr|pmn⟩     \textcelsius              = { 50,   },
⟨pad⟩     \textcelsius              = { 80,   },
⟨bch⟩     \textflorin               = { 50, 50},
⟨pad⟩     \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},
⟨bch⟩       \textcent                 = { 50,   },
⟨ptm⟩       \textcent                 = {100,100},
⟨bch⟩       \textsterling             = { 50,   },
⟨bch⟩       \textbrokenbar            = {200,200},
⟨pmn⟩       \textasciidieresis        = {300,400},
⟨m-t|bch|cmr|pad|ptm⟩       \textcopyright            = {100,100},
⟨pmn⟩       \textcopyright            = {100,150},
⟨ppl⟩       \textcopyright            = {200,200},
⟨bch|cmr⟩       \textordfeminine          = {100,200},
⟨pad|pmn⟩       \textordfeminine          = {200,200},
⟨bch|cmr|pad|pmn⟩       \textlnot                 = {200,   },
⟨m-t|bch|cmr|pad|ptm⟩       \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},
⟨bch|cmr|pad|pmn⟩       \textpm                   = {150,200},
⟨ptm⟩       \textpm                   = { 50, 80},
⟨bch⟩       \texttwosuperior          = {100,200},
⟨cmr⟩       \texttwosuperior          = { 50,100},
⟨pad|pmn⟩       \texttwosuperior          = {200,200},
⟨ptm⟩       \texttwosuperior          = { 50, 50},
⟨bch⟩       \textthreesuperior        = {100,200},
⟨cmr⟩       \textthreesuperior        = { 50,100},
⟨pad|pmn⟩       \textthreesuperior        = {200,200},
⟨ptm⟩       \textthreesuperior        = { 50, 50},
⟨pmn⟩       \textasciiacute           = {300,400},
⟨bch⟩       \textmu                   = {   ,100},
⟨bch|pad|pmn⟩       \textparagraph            = {   ,100},
⟨bch|cmr|pad|pmn⟩       \textperiodcentered       = {300,400},
⟨ptm⟩       \textperiodcentered       = {300,300},
⟨bch⟩       \textonesuperior          = {200,300},
⟨cmr|pad|pmn⟩       \textonesuperior          = {200,200},
⟨ptm⟩       \textonesuperior          = {100,100},
⟨bch|pad|pmn⟩       \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},
⟨bch|pad|pmn⟩       \textdiv                  = {150,200},
⟨cmr⟩       \textdiv                  = {150,250},
⟨ptm⟩       \textdiv                  = { 50,100},
⟨ptm⟩       \textperthousand          = {   ,50},
```

All remaining characters can be found in the source.

```
   }

⟨*cmr|pad|pmn⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-textcomp-it ]
⟨pad⟩   [ name     = pad-textcomp-it ]
```

```
⟨pmn⟩   [ name     = pmn-textcomp-it ]
   { encoding = TS1,
⟨cmr⟩     family   = cmr,
⟨pad⟩     family   = {pad,padx,padj},
⟨pmn⟩     family   = {pmnx,pmnj},
     shape    = {it,sl} }
   {
⟨cmr⟩     \textquotestraightbase    = {300,600},
⟨pad|pmn⟩     \textquotestraightbase    = {400,400},
⟨cmr⟩     \textquotestraightdblbase = {300,600},
⟨pad⟩     \textquotestraightdblbase = {300,400},
⟨pmn⟩     \textquotestraightdblbase = {300,300},
     \texttwelveudash          = {200,200},
     \textthreequartersemdash  = {150,150},
⟨cmr⟩     \textquotesingle          = {600,300},
⟨pad⟩     \textquotesingle          = {800,100},
⟨pmn⟩     \textquotesingle          = {300,200},
⟨cmr⟩     \textasteriskcentered     = {300,200},
⟨pad⟩     \textasteriskcentered     = {500,100},
⟨pmn⟩     \textasteriskcentered     = {200,300},
⟨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},
⟨pad|pmn⟩     \textlbrackdbl            = {100,   },
⟨pad|pmn⟩     \textrbrackdbl            = {   ,100},
⟨pmn⟩     \textasciigrave           = {300,300},
     \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},
⟨cmr|pad⟩     \textdaggerdbl            = { 80, 80},
⟨pmn⟩     \textdaggerdbl            = { 80, 50},
⟨cmr⟩     \textbullet               = {200,100},
⟨pad⟩     \textbullet               = {300,   },
⟨pmn⟩     \textbullet               = { 30, 70},
⟨cmr⟩     \textcelsius              = {100,   },
⟨pad⟩     \textcelsius              = {200,   },
⟨pmn⟩     \textcelsius              = { 50,-50},
⟨pad⟩     \textflorin               = {100,   },
⟨pmn⟩     \textflorin               = { 50,100},
⟨cmr⟩     \textcolonmonetary        = {150,   },
⟨pad⟩     \textcolonmonetary        = {100,   },
⟨pmn⟩     \textcolonmonetary        = { 50,-50},
⟨cmr|pad⟩     \texttrademark            = {200,   },
⟨pmn⟩     \texttrademark            = { 50,100},
⟨pmn⟩     \textasciidieresis        = {300,200},
⟨cmr⟩     \textcopyright            = {100,   },
⟨pad⟩     \textcopyright            = {200,100},
⟨pmn⟩     \textcopyright            = {100,150},
⟨cmr⟩     \textordfeminine          = {100,100},
⟨pmn⟩     \textordfeminine          = {200,200},
⟨cmr|pad⟩     \textlnot                 = {300,   },
⟨pmn⟩     \textlnot                 = {200,   },
⟨cmr⟩     \textregistered           = {100,   },
⟨pad⟩     \textregistered           = {200,100},
⟨pmn⟩     \textregistered           = { 50,150},
⟨pmn⟩     \textasciimacron          = {150,200},
⟨cmr|pad⟩     \textdegree               = {500,100},
⟨pmn⟩     \textdegree               = {150,150},
⟨cmr⟩     \textpm                   = {150,100},
⟨pad⟩     \textpm                   = {200,150},
⟨pmn⟩     \textpm                   = {150,200},
⟨cmr⟩     \textonesuperior          = {400,   },
⟨pad⟩     \textonesuperior          = {300,100},
⟨pmn⟩     \textonesuperior          = {200,100},
⟨cmr⟩     \texttwosuperior          = {400,   },
⟨pad⟩     \texttwosuperior          = {300,   },
⟨pmn⟩     \texttwosuperior          = {200,100},
⟨cmr⟩     \textthreesuperior        = {400,   },
⟨pad⟩     \textthreesuperior        = {300,   },
⟨pmn⟩     \textthreesuperior        = {200,100},
⟨pmn⟩     \textasciiacute           = {300,200},
⟨cmr⟩     \textparagraph            = {200,   },
⟨pmn⟩     \textparagraph            = {   ,100},
⟨cmr⟩     \textperiodcentered       = {500,500},
⟨pad|pmn⟩     \textperiodcentered       = {300,400},
⟨cmr⟩     \textordmasculine         = {100,100},
⟨pmn⟩     \textordmasculine         = {200,200},
⟨cmr⟩     \texteuro                 = {200,   },
⟨pad⟩     \texteuro                 = {100,   },
⟨pmn⟩     \texteuro                 = {100,-50},
⟨cmr⟩     \texttimes                = {200,200},
⟨pad⟩     \texttimes                = {200,100},
⟨pmn⟩     \texttimes                = { 70,100},
⟨cmr|pad⟩     \textdiv                  = {200,200},
⟨pmn⟩     \textdiv                  = {150,200},
    }

⟨/cmr|pad|pmn⟩
```

14.6.6 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
%    "11 = {   ,   }, % \eta
     "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
%    "1E = {   ,   }, % \phi
     "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
     % 0 - 9
     "3A = {   ,500}, % ., \ldotp
     "3B = {   ,500}, % ,
     "3C = {200,100}, % <
     "3D = {300,400}, % /
     "3E = {100,200}, % >
     "3F = {200,200}, % \star
%    "40 = {   ,   }, % \partial
     "5B = {   ,100}, % \flat
%    "5C = {   ,   }, % \natural
%    "5D = {   ,   }, % \sharp
     "5E = {200,200}, % \smile
     "5F = {200,200}, % \frown
%    "60 = {   ,   }, % \ell
%    "7B = {   ,   }, % \imath
     "7C = {100,   }, % \jmath
     "7D = {   ,100}, % \wp
   }

```

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
%   "36 = {   ,   }, % \not
%   "37 = {   ,   }, % \mapstochar
     "38 = {   ,100}, % \forall
     "39 = {100,   }, % \exists
     "3A = {200,   }, % \neg
%   "3B = {   ,   }, % \emptyset
%   "3C = {   ,   }, % \Re
%   "3D = {   ,   }, % \Im
     "3E = {200,200}, % \top
     "3F = {200,200}, % \bot, \perp
%   "40 = {   ,   }, % \aleph
%   "5B = {   ,   }, % \cup
%   "5C = {   ,   }, % \cap
%   "5D = {   ,   }, % \uplus
     "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
%   "6A = {   ,   }, % \arrowvert, \mid, \vert, |
%   "6B = {   ,   }, % \Arrowvert, \parallel, \Vert
     "6C = {100,100}, % \updownarrow
     "6D = {100,100}, % \Updownarrow
     "6E = {100,300}, % \, \backslash, \setminus
%   "6F = {   ,   }, % \wr
%   "70 = {   ,   }, % \sqrtsign
%   "71 = {   ,   }, % \amalg
     "72 = {100,100}, % \nabla
%   "73 = {   ,   }, % \smallint
%   "74 = {   ,   }, % \sqcup
%   "75 = {   ,   }, % \sqcap
%   "76 = {   ,   }, % \sqsubseteq
%   "77 = {   ,   }, % \sqsupseteq
%   "78 = {   ,   }, % \mathsection
     "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
   }

```

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}
```

```
%\SetProtrusion
%   [ name     = cmr-math-largesymbols ]
%   { encoding = OMX,
%     family   = {cmex,cmr} }
%   {
%     "00 % (
%     "01 % )
%     "02 % [
%     "03 % ]
%     "04 % \lfloor
%     "05 % \rfloor
%     "06 % \lceil
%     "07 % \rceil
%     "08 % \lbrace
%     "09 % \rbrace
%     "0A % <, \langle
%     "0B % >, \rangle
%     "0C % \vert, |
%     "0D % \Vert
%     "0E % /
%     "0F % \backslash
%     "3A % \lgroup
%     "3B % \rgroup
%     "3C % \arrowvert
%     "3D % \Arrowvert
%     "3E % \bracevert
%     "3F % \updownarrow
%     "40 % \lmoustache
%     "41 % \rmoustache
%     "46 % \bigsqcup
%     "48 % \ointop
%     "4A % \bigodot
%     "4C % \bigoplus
%     "4E % \bigotimes
%     "50 % \sum
%     "51 % \prod
%     "52 % \intop
%     "53 % \bigcup
%     "54 % \bigcap
%     "55 % \biguplus
%     "56 % \bigwedge
%     "57 % \bigvee
%     "60 % \coprod
%     "70 % \sqrtsign
%     "77 % \Updownarrow
%     "78 % \uparrow
%     "79 % \downarrow
%     "7A % \braceld, \lmoustache
%     "7B % \bracerd, \rmoustache
%     "7C % \bracelu
%     "7D % \braceru
%     "7E % \Uparrow
%     "7F % \Downarrow
%  }
⟨/cmr⟩
```

⟨/!cfg-u⟩

14.6.7 AMS fonts

*⟨*cfg-u⟩*

Symbol font ‘a’, defined in `amssymb`.

```
\DeclareSymbolFont{AMSa}{U}{msa}{m}{n}
```

*⟨*msa⟩*

```
\SetProtrusion
   [ name     = AMSa ]
   { encoding = U,
     family   = msa  }
   {
%    "00  =  {   ,   },  % \boxdot
%    "01  =  {   ,   },  % \boxplus
%    "02  =  {   ,   },  % \boxtimes
%    "03  =  {   ,   },  % \square
%    "04  =  {   ,   },  % \blacksquare
     "05  =  {150,250},  % \centerdot
     "06  =  {100,100},  % \lozenge
     "07  =  { 50, 50},  % \blacklozenge
     "08  =  { 50, 50},  % \circlearrowright
     "09  =  { 50, 50},  % \circlearrowleft
     "0A  =  { 50, 50},  % \rightleftharpoons
     "0B  =  { 50, 50},  % \leftrightharpoons
%    "0C  =  {   ,   },  % \boxminus
     "0D  =  {-50,150},  % \Vdash
     "0E  =  {-50,150},  % \Vvdash
     "0F  =  {-70,150},  % \vDash
     "10  =  {100,100},  % \twoheadrightarrow
     "11  =  { 50,150},  % \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  =  { 50, 80},  % \rightarrowtail
     "1B  =  { 50, 80},  % \leftarrowtail
     "1C  =  { 50, 50},  % \leftrightarrows
     "1D  =  { 50, 50},  % \rightleftarrows
     "1E  =  {150,   },  % \Lsh
     "1F  =  {   ,150},  % \Rsh
     "20  =  { 50,100},  % \rightsquigarrow
     "21  =  { 50, 50},  % \leftrightsquigarrow
     "22  =  { 50, 50},  % \looparrowleft
     "23  =  { 50, 50},  % \looparrowright
     "24  =  {   , 50},  % \circeq
     "25  =  {   ,100},  % \succsim
     "26  =  {   ,100},  % \gtrsim
     "27  =  {   ,100},  % \gtrapprox
     "28  =  { 50, 50},  % \multimap
%    "29  =  {   ,   },  % \therefore
%    "2A  =  {   ,   },  % \because
     "2B  =  { 50, 50},  % \doteqdot
```

```
    "2C = { 50, 50}, % \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
%   "35 = {   ,   }, % \leqq
    "36 = { 50,   }, % \leqslant
%   "37 = {   ,   }, % \lessgtr
    "38 = {   , 50}, % \backprime
    "39 = {200,200}, % \dabar@ : the dash bar in \dash(left,right)arrow
%   "3A = {   ,   }, % \risingdotseq
%   "3B = {   ,   }, % \fallingdotseq
    "3C = { 50,100}, % \succcurlyeq
%   "3D = {   ,   }, % \geqq
    "3E = {   , 50}, % \geqslant
%   "3F = {   ,   }, % \gtrless
    "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
%   "47 = {   ,   }, % \between
    "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
%   "51 = {   ,   }, % \lesseqgtr
%   "52 = {   ,   }, % \gtreqless
%   "53 = {   ,   }, % \lesseqqgtr
%   "54 = {   ,   }, % \gtreqqless
%   "55 = {   ,   }, % \yen
    "56 = {   ,150}, % \Rrightarrow
    "57 = {150,   }, % \Lleftarrow
    "58 = {100,200}, % \checkmark
%   "59 = {   ,   }, % \veebar
%   "5A = {   ,   }, % \barwedge
%   "5B = {   ,   }, % \doublebarwedge
    "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
%   "64 = {   ,   }, % \Cup
%   "65 = {   ,   }, % \Cap
    "66 = {100,100}, % \curlywedge
    "67 = {100,100}, % \curlyvee
```

```
     "68  =  { 50,100},  % \leftthreetimes
     "69  =  {100, 50},  % \rightthreetimes
%    "6A  =  {   ,   },  % \subseteqq
%    "6B  =  {   ,   },  % \supseteqq
     "6C  =  { 50, 50},  % \bumpeq
     "6D  =  { 50, 50},  % \Bumpeq
     "6E  =  {100,   },  % \lll
     "6F  =  {   ,100},  % \ggg
     "70  =  { 50,100},  % \ulcorner
     "71  =  {100, 50},  % \urcorner
%    "72  =  {   ,   },  % \circledR
%    "73  =  {   ,   },  % \circledS
%    "74  =  {   ,   },  % \pitchfork
     "75  =  {100,150},  % \dotplus
     "76  =  { 50,100},  % \backsim
%    "77  =  {   ,   },  % \backsimeq
     "78  =  { 50,100},  % \llcorner
     "79  =  {100, 50},  % \lrcorner
%    "7A  =  {   ,   },  % \maltese
%    "7B  =  {   ,   },  % \complement
     "7C  =  {100,100},  % \intercal
     "7D  =  { 50, 50},  % \circledcirc
     "7E  =  { 50, 50},  % \circledast
     "7F  =  { 50, 50},  % \circleddash
   }

⟨/msa⟩
```

Symbol font ‘b’.

```
\DeclareSymbolFont{AMSb}{U}{msb}{m}{n}
\DeclareSymbolFontAlphabet{\mathbb}{AMSb}
```

```
⟨*msb⟩
\SetProtrusion
   [ name      = AMSb ]
   { 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
%    "40  =  {   ,   },  % \nexists
     "60  =  {200,   },  % \Finv
     "61  =  {   , 50},  % \Game
%    "66  =  {   ,   },  % \mho
%    "67  =  {   ,   },  % \eth
     "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
%    "7C  =  {   ,   },  % \Bbbk
%    "7D  =  {   ,   },  % \hslash
%    "7E  =  {   ,   },  % \hbar
     "7F  =  {200,   },  % \backepsilon
   }

⟨/msb⟩
```

Euler Fraktur font (`eufrak`).

```
\DeclareMathAlphabet{\mathfrak}{U}{euf}{m}{n}
\SetMathAlphabet{\mathfrak}{bold}{U}{euf}{b}{n}
```

```
⟨*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},
   }
```

```

⟨/euf⟩
```

Euler script font (`eucal`).

```
\DeclareMathAlphabet\EuScript{U}{eus}{m}{n}
\SetMathAlphabet\EuScript{bold}{U}{eus}{b}{n}
```

```
⟨*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},
       Z  =  { 50,100},
   }

⟨/eus⟩
⟨/cfg-u⟩
⟨*beta⟩
```

14.7 Interword Spacing

Default unit is space.

```
%%% ----------------------------------------------------------------------
%%% INTERWORD SPACING SETTINGS

\SetExtraSpacing
   [ name = default ]
   { encoding = {OT1,T1,LY1,OT4,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.

> ‘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.)

The following settings simulate `\nonfrenchspacing` (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the TEXbook:

> ‘If the space factor *f* is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f\,/\,1000$, while the shrink component is multiplied by $1000\,/\,f$.’

The ‘extra space’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen` 2, i. e., 333.

```
\SetExtraSpacing
  [ name     = nonfrench-cmr,
    load     = default,
    context  = nonfrench ]
  { encoding = {OT1,T1,LY1,OT4,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 it 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,T5} }
  {
    . = {240,2000,-667},
    ? = {240,2000,-667},
    ! = {240,2000,-667},
    : = {240,1000,-500},
    ; = {   , 500,-333},
   {,}= {   , 250,-200},
  }

```

14.8 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,T5,TS1} }
  { }

\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 guillemot will not be hyphenated. This might be fixed in pdfTEX.

```
\SetExtraKerning
  [ name     = french-guillemets,
    context  = french-guillemets,
    load     = french-default,
    unit     = space   ]
  { encoding = {OT1,T1,LY1} }
  {
   \guillemotleft  = { ,800}, % = 0.8\fontdimen2
   \guillemotright = {800, },
  }

```

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

```

The settings with the ‘`letterspacing`’ context will be loaded whenever the command `\textls` resp. `\lsstyle` are used.

```
%%% The following settings with 'context=letterspacing'
%%% will apply to \lsstyle and \textls:
\SetExtraKerning
   [ name     = letterspacing-default,
     context  = letterspacing,
     unit     = 1em,
     preset   = {1000,1000} ]
   { encoding = {OT1,OT4}   }
   {
```

The full stop and quotation marks should be spaced out less. Numbers are not spaced out, according to `soul`.

```
     .  = {0, },
     0  = {0,0},
     1  = {0,0},
     2  = {0,0},
     3  = {0,0},
     4  = {0,0},
     5  = {0,0},
     6  = {0,0},
     7  = {0,0},
     8  = {0,0},
     9  = {0,0},
     \textquoteleft    = {0,0},   \textquoteright    = {0,0},
     \textquotedblleft = {0,0},   \textquotedblright = {0,0},
   }

\SetExtraKerning
   [ name     = letterspacing-T1,
     load     = letterspacing-default,
     context  = letterspacing,
     unit     = 1em,
     preset   = {1000,1000} ]
   { encoding = {T1,LY1,T5} }
   {
     \quotesinglbase   = {0,0},   \quotedblbase      = {0,0},
     \guilsinglleft    = {0,0},   \guilsinglright    = {0,0},
     \guillemotleft    = {0,0},   \guillemotright    = {0,0},
   }

⟨/beta⟩
⟨/config⟩
```

15 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) 57
  issue an error instead of a warning, when pdfTEX version is too old for `autoexpand` 102
  remove 8-bit characters from the configuration files (suggested by Harald Harders) 108
Protrusion: add factors for some more characters 112
  settings for Adobe Minion (contributed by Harald Harders) . . . 113
`\DeclareCharacterInheritance`: new macro: possibility to specify character inheritance 85
`\MT@declare@sets`: remove spaces around set name 72
`\MT@DeclareSet@`: remove spaces around first argument 71
`\MT@find@file`: bug fix: also check whether the file for the base font family has already been loaded . 57
`\MT@get@basefamily`: only remove suffix, if it is ‘x’ or ‘j’ 58
`\MT@get@listname@`: don’t check for empty characteristics list 58
`\MT@ifempty`: bug fix: use category code 12 for the percent character (reported by Tom Kink) 34
`\MT@is@number`: numbers may also be specified in hexadecimal or octal (suggested by Harald Harders) . . 62
`\MT@pdftex@no`: bug fix concerning version check (reported by Harald Harders) 26
`\MT@permute`: don’t use sets for empty encoding 88
`\MT@pr@split`: bug fix: allow zero and negative values 43
`\MT@use@set`: remove spaces around set name 76
`\UseMicrotypeSet`: remove spaces around first argument 76

Version 1.2 (2004/10/03)

General: check for packages that might load fonts 66
Font Sets: declare `cmor` as an alias of `cmr` 107
  new: `allmath` and `basicmath` . . . 106
Protrusion: add settings for Adobe Garamond and Computer Modern Roman in TS1 encoding 130
  add settings for Computer Modern Roman math symbols 134
`\MT@context`: bug fix: set inheritance list `\globally` to `\@empty` 60
`\MT@define@inh@key@encoding`: check whether only one encoding specified 86
`\MT@familyalias`: define alias font name as an alternative, not as a replacement 40
`\MT@get@basefamily`: also remove ‘w’ (swash capitals) 58
`\MT@get@highlevel`: check whether defaults have changed 73
`\MT@get@listname@`: alternatively check for alias font name 58
`\MT@get@size`: additional magic to catch some errors 74
  hijack `\set@fontsize` instead of `\@setfontsize` 74
`\MT@get@slot`: bug fix: group must also include `\MT@get@composite` 61
`\MT@loop`: bug fix: new macro, used instead of `\loop` 37

Version 1.3 (2004/10/27)

Version 1.4 (2004/11/12)

Version 1.4a (2004/11/17)

Version 1.4b (2004/11/26)

Version 1.5 (2004/12/15)

- General: defaults: `step`: 4 (suggested by Hàn Thế Thành) . . . 97
  - defaults: calculate `step` as `min(stretch,shrink)/5` . . . 101
  - defaults: turn off expansion for DVI output . . . 100
  - disable automatic expansion for DVI output . . . 102
  - new option: `selected`, by default false (suggested by Hàn Thế Thành) . . . 95
- Documentation: add note about `DVIoutput` option . . . 7
  - add short history (section 12) . . . 21
- Inheritance: remove `\ss` from T1 list, add `\DJ` . . . 108
- Protrusion: settings for Bitstream Charter . . . 113
- `\DeclareMicrotypeAlias`: remove spaces around arguments . . . 77
- `\MT@begin@catcodes`: reset catcode of ‘=’ (compatibility with Turkish `babel`) . . . 58
- `\MT@catcodes`: reset catcode of ‘^’ (compatibility with `chemsym`) . . . 27
- `\MT@get@highlevel`: don’t test defaults if called after begin document . . . 73
- `\MT@scale@factor`: warning for factors outside limits . . . 45
- `\MT@scale@to@em`: don’t use `\lpcode` and `\rpcode` for the calculation . . . 44
- `\MT@set@ex@codes`: allow non-selected font expansion . . . 49
- `\MT@set@pr@codes`: adjust protrusion factors before setting the inheriting characters . . . 42

Version 1.6 (2005/01/24)

- General: defaults: turn off expansion for old pdfTeX versions . . . 98
  - disable automatic expansion for old pdfTeX versions . . . 102
  - load a font, if none is active . . . 32
  - new option: `factor`, by default 1000 . . . 97
  - restructure `dtx` file . . . 106
  - test whether `\pickup@font` has changed . . . 68
  - use e-TeX’s `\ifcsname` and `\ifdefined` if defined . . . 32
- Protrusion: add italic uppercase Greek letters . . . 119
  - improve settings for numbers (pointed out by Peter Muthesius) . . . 114
  - tune CMR math letters (OML encoding) . . . 135
- `\MT@def@num@opt`: test whether numeric options receive a number . . . 97
- `\MT@get@charwd`: use e-TeX’s `\fontcharwd`, if available . . . 44
- `\MT@get@inh@list`: correct message if `selected` is false . . . 60
- `\MT@set@ex@codes`: introduce `factor` option . . . 49
- `\MT@set@pr@codes`: introduce `factor` option . . . 42
- `\MT@use@set`: retain current set if new set is undeclared . . . 76
- `\MT@vinfo`: new macro: used instead of `\ifMT@verbose` . . . 25

Version 1.6a (2005/02/02)

- Documentation: add table of fonts with tailored protrusion settings . . . 15
- `\MT@get@slot`: completely redone, hopefully more robust (compatible with `frenchpro`; problem reported by Bernard Gaulle) . . . 61
- `\MT@pdftex@no`: new macro . . . 26
- `\MT@reset@ef@codes`: only reset `\efcodes` for older pdfTeX versions . . . 49

Version 1.7 (2005/03/23)

- General: `\SetExpansion`: bug fix: remove space after `autoexpand` 85
  - `\SetExpansion`: don't allow automatic expansion for old pdfTEX versions 85
  - allow specification of size ranges (suggested by Andreas Bühmann) 73
  - modify `\showhyphens` 103
  - new value for `verbose` option: `errors` 96
  - shorter command names 38
  - warning when running in draft mode 100
- Documentation: add hint about compatibility 18
  - remove table of match order 12
- Protrusion: fix: remove \ from OT1, add `\textbackslash` to T1 encoding 115
- `\DeclareMicrotypeAlias`: may also be used inside configuration files 77
- `\LoadMicrotypeFile`: new macro (suggested by Andreas Bühmann) 78
- `\Microtype@Hook`: new macro for font package authors 98
- `\microtypesetup`: bug fix: warning also when setting to `(no)compatibility` 91
- `\MT@begin@catcodes`: also use inside configuration commands 58
  - reset catcode of ':' (compatibility with `french*` packages) 58
- `\MT@get@listname@`: use `\@tfor` (Andreas Bühmann's idea) 58
- `\MT@get@size`: comparison with 1 to allow size smaller than 1 (suggested by Andreas Bühmann) 74
- `\MT@get@slot`: remove backslash hack 61
  - test for `\chardefed` commands 61
  - test whether \⟨*encoding*⟩\⟨`..`⟩ is defined 61
- `\MT@if@list@exists`: don't define `\MT@#1@c@name` `\globally`, here and elsewhere 60
- `\MT@increment`: use e-TEX's `\numexpr` if available 38
- `\MT@is@composite`: new macro: construct command for composite character; no uncontrolled expansion 65
- `\MT@scale`: new macro: use e-TEX's `\numexpr` if available 38
- `\MT@set@ex@codes`: two versions of this macro 49
- `\MT@split@name`: don't define `\MT@encoding` &c. `\globally` 40
- `\MT@test@ast`: make it simpler 73
- `\MT@try@order`: always check for size, too (suggested by Andreas Bühmann) 59
  - bug fix: also check for //⟨*series*⟩/⟨*shape*⟩// (reported by Andreas Bühmann) 59
- `\MT@warn@code@too@large`: new macro: type out maximum protrusion factor 46
- `\MT@warn@err`: new macro: for `verbose=errors` 25

Version 1.8 (2005/06/23)

- General: `\SetProtrusion`: new option: `unit` 84
  - if font substitution has occurred, set up the substitute font, not the selected one 67
  - new option: `config` to load a different main configuration file 98
  - new option: `unit`, by default `character` 97
- Documentation: add example for `factor` option 12
  - add example of how to get rid of a widow 13
  - add hint about error messages 19
- Font Sets: add U encoding to `allmath` 106
  - declare `pxr` and `txr` as alias fonts of `ppl` and `ptm` 107
- Inheritance: remove `\DJ` from T1 list (it's the same as `\DH`) 108
- Protrusion: add LY1 characters for Times 118
  - settings for AMS math fonts 113
  - verified settings for slanted Computer Modern Roman 124

- `\DeclareMicrotypeAlias`: warning when overriding an alias font . . . 77
- `\DeclareMicrotypeSetDefault`: new macro: set fall back font set 77
- `\MT@begin@catcodes`: reset catcodes of the remaining ASCII characters 58
- `\MT@check@rlist`: made recursive 90
- `\MT@curr@list@name`: new macro: current list type and name 65
- `\MT@declare@sets`: warning when redefining a set 72
- `\MT@define@set@key@`: use comma lists instead of token lists 72
- `\MT@find@file`: no longer wrap names in commands 57
- `\MT@get@charwd`: warning for missing (resp. zero-width) characters . . . 44
- `\MT@get@dimen@six`: new macro: test whether `\fontdimen6` is defined 43
- `\MT@get@listname@`: made recursive . . 58
- `\MT@get@slot`: bug fix: expand active characters 61
  - test whether `\`⟨*encoding*⟩`\`⟨`..`⟩ is defined made more robust 61
- `\MT@get@unit`: new macro: get unit for codes 47
- `\MT@in@rlist`: made recursive 37
- `\MT@is@active`: new macro: translate `inputenc`-defined characters . . . 63
- `\MT@is@letter`: warning for non-ASCII characters 62
- `\MT@led@kern`: character protrusion with `ledmac` 29
- `\MT@make@string`: use `\@onelevel@sanitize` 38
- `\MT@map@clist@n`: new macro: used instead of `\@for` 35
- `\MT@map@tlist@n`: new macro: used instead of `\@tfor` 36
- `\MT@old@cmd`: renamed commands from `\..MicroType..` to `\..Microtype..` 25
- `\MT@orig@add@accent`: bug fix: disable micro-typographic setup inside `\add@accent` (reported by Stephan Hennig) 68
- `\MT@pdftex@no`: case 5: pdfTeX 1.30 . . 26
- `\MT@permute@@@@@@`: add ranges to the beginning of the lists 89
- `\MT@pr@split`: get character width once only 43
- `\MT@scale`: bug fix: remove spaces in non-e-TeX variant (reported by Mark Rossi) 38
- `\MT@setupfont@hook`: restore `\%` and `\#` when `hyperref` is loaded 31
  - restore `csquotes`'s active characters 30
  - restore percent character if Spanish `babel` is loaded 30
- `\MT@use@set`: bug fix: remove braces in first line 76
- `\MT@xadd`: simplified 35

Version 1.9 (2005/10/28)

- General: `\DeclareMicrotypeSet`: new key: `font` 75
  - `\SetProtrusion` and `\SetExpansion`: new key: `font` 82
  - `\SetProtrusion`: value '`relative`' renamed to '`character`' for option `unit` 84
  - allow context-specific font setup . . 67
  - disable expansion if both `step` and `shrink` are zero 102
  - disable `microtype` setup inside `hyperref`'s `\pdfstringdef` (reported by Hàn Thế Thành) 31
  - option `unit`: rename value `relative` to `character` 97
  - warning if user requested zero `step` 101
- Documentation: add hint about `verbatim` environment 18
  - add remark about Type 1 fonts required for automatic font expansion 6
- Font Sets: add OT4 encoding to text sets 106
  - add T5 encoding to text sets 106
  - declare `qpl` and `qtm` (`qfonts`) as alias fonts of `ppl` resp. `ptm` 107
- Inheritance: add list for OT4 110
  - add list for T5 (requested by Hàn Thế Thành) 110
- Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR . 115
  - settings for OT4 encoding (Computer Modern Roman, Palatino, Times) 112

settings for T5 encoding (Computer Modern Roman) 112
`\DisableLigatures`: new macro: disable ligatures (requires pdfTeX 1.30) 78
`\microtypecontext`: new macro: change setup context in the document . . 69
`\MT@checklist@family`: bug fix: add two missing `\expandafter`s 41
`\MT@define@option`: fix: use `true` as the default value 94
`\MT@detokenize`: fix the non-e-TeX version 65
`\MT@exp@two@n`: new macros: less `\expandafter`s 32
`\MT@get@opt`: new key 'preset' to set all characters to the specified value before loading the lists 47
`\MT@is@active`: redone: use `\set@display@protect` 63
`\MT@is@letter`: using `\catcode` should be more efficient than inspecting the `\meaning` 62
`\MT@maybe@do`: redone 40
`\MT@pdftex@no`: compatibility with TeXLive hack (reported by Herbert Voß) 26
`\MT@rem@from@clist`: new macro: remove an item from a comma list 36
`\MT@scale@factor`: generalized 45
`\MT@toks`: no longer use `\toks@` 28

Version 1.9a (2005/12/05)

General: new option: `defersetup`, by default true 96
remove superfluous test whether `\pickup@font` has changed 68
Documentation: add explanation for error message in DVI mode 19
add explanation for error message with bitmap fonts 19
Font Sets: declare `mdbch` (`mathdesign`) as an alias of Charter 107
Protrusion: settings for T5 encoded Charter 112
`\microtypesetup`: inside the preamble, accepts all package options 91
`\MT@check@font@cx`: optimize context-sensitive setup 68
`\MT@define@set@key@`: don't expand variables immediately (requested by Georg Verweyen) 72
`\MT@get@codes@name`: 'file name/line number' as default list name . . . 81
`\MT@get@highlevel`: no longer check whether defaults have changed 73
`\MT@ifdefined@c@T`: new macros: TRUE case only 33
`\MT@ifnumber`: use `\pdfmatch` if available 34
`\MT@ifstreq`: use `\pdfstrcmp` if available 35
`\MT@in@clist`: fix 36
`\MT@info@missing@char`: info instead of warning (after Michael Hoppe reported that the 'fl' ligature is missing in Palatino SC) 45
`\MT@is@feature`: new macro: check for pdfTeX feature 71
`\MT@map@clist@n`: following LaTeX3 . . . 35
`\MT@permute@@@@@`: don't define permutations for unused encodings . . 88
`\MT@rem@from@clist`: fix 36
`\MT@setup@microtype`: defer setup until the end of the preamble 31

Version 2.0 ()

General: new option: `babel`, by default false (language-dependant setup suggested by Ulrich Dirr) 95
new option: `letterspacing`, by default 100 97
`\DeclareMicrotypeBabelHook`: new command: interaction with `babel` 78
`\MT@pdftex@no`: case 6: pdfTeX 1.4x . . 26
`\SetExtraKerning`: new macro: additional kerning 80
`\SetExtraSpacing`: new macro: adjustment of interword spacing 80
`\textls`: new command: letterspacing 56

B Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.