The microtype package

Subliminal refinements towards typographical perfection

R Schlicht
w.m.l@gmx.net

v2.7b
2019/02/28

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

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

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

This package is copyright © 2004–2018 R Schlicht. It may be distributed and/or modified under the conditions of the \TeX{} Project Public License, either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.
Contents

1 Micro-typography with TeX 4
2 Getting started 5
3 Options 6
   3.1 Enabling the micro-typographic features 6
   3.2 Character protrusion 7
   3.3 Font expansion 7
   3.4 Tracking 8
   3.5 Miscellaneous options 8
   3.6 Changing options later 9
4 Selecting fonts for micro-typography 10
5 Micro fine tuning 12
   5.1 Character protrusion 13
   5.2 Font expansion 14
   5.3 Tracking 15
   5.4 Additional kerning 18
   5.5 Interword spacing 19
   5.6 Character inheritance 20
   5.7 Configuration files 20
6 Context-sensitive setup 22
7 Letterspacing revisited 23
8 Disabling ligatures 24
9 Hints and caveats 25
10 Contributions 28
11 Acknowledgments 28
12 References 29
13 Short history 30
14 Implementation 34
   14.1 Preliminaries 35
   14.2 Font setup 56
   14.3 Configuration 103
   Compatibility [51]
   Protrusion [61] Expansion [68] Interword spacing (glue) [71] Additional kerning [72]
   Tracking [74] Disabling ligatures [84] Loading the configuration [86]
   Translating characters into slots [90] Hook into \TeX’s font selection [97]
   Context-sensitive setup [100]
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Availability of micro-typographic features</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Predefined font sets</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Fonts with tailored protrusion settings</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Order for matching font attributes</td>
<td>88</td>
</tr>
</tbody>
</table>
1 Micro-typography with \TeX

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

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

Another micro-typographic technique, which has always been extremely difficult to achieve in \TeX{}, is robust and hyphenatable letterspacing (tracking).\footnote{The soul package undertakes great efforts, but may still fail in certain circumstances; even to systematically adjust the tracking of a font throughout the document remains impossible.} Whereas letterspacing can easily be, and often is, abused when applying it to lowercase letters, readability may be increased by slightly letterspacing (small) capitals or by decreasing the tracking of very large uppercase type.

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

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

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

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

2 Getting started

There is nothing surprising in loading this package:

\begin{verbatim}
\usepackage{microtype}
\end{verbatim}

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

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

Like many other \LaTeX\ packages, the \texttt{microtype} package accepts options in the well-known \texttt{key=value} syntax. In the following, you will find a description of all \texttt{keys} and their possible \texttt{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 \TeX\ engine, version and/or the output mode).

3.1 Enabling the micro-typographic features

- **protrusion** \texttt{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 always be enabled, font expansion will only be disabled when the fonts cannot be expanded automatically, that is, with pdf\TeX\ versions older than 1.20 or in DVI output mode (see section 3.5), or with X\TeX. In other words, \texttt{microtype} will try to apply as much micro-typography as can safely be expected to work under the respective conditions (hence, it is usually not necessary to load the package with different options for PDF resp. DVI mode).

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

\begin{verbatim}
\usepackage[protrusion=true,expansion]{microtype}
\usepackage[activate={true,nocompatibility}]{microtype}
\usepackage{microtype}
\end{verbatim}

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

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

- **activate** \texttt{true, false, (font set name)} \texttt{false}

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

- **kerning** \texttt{true, false, (font set name)} \texttt{false}

These features do not unconditionally improve the quality of the typeset text: the \texttt{kerning} feature is still considered experimental, while the \texttt{spacing} feature only makes sense in special cases. Therefore, neither feature is enabled by default. They are not available with X\TeX\ or Lua\TeX.
Table 1 presents an overview of which micro-typographic features are available and enabled by default for the relevant \TeX versions and output modes.

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

3.2 Character protrusion

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).

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

3.3 Font expansion

Beginning with \pdfTeX version 1.20 (inherited by \LaTeXX), the expanded instances of the fonts may be calculated automatically and at run-time instead of the user having to prepare them in advance. This option is true by default provided that you are using a \TeX engine with this capability and the output mode is PDF. If auto...
is set to false, the font instances for all expansion steps must exist (with files called \textit{font name}\texttt{±}\textit{expansion value}, e.g., \texttt{cmr12±10}, as described in the \texttt{pdfTeX manual}).

With \texttt{pdfTeX}, automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in \texttt{T1} encoding, you should either install the \texttt{cm-super} fonts or use the Latin Modern fonts (package \texttt{lmodern}). With \texttt{LuaTeX}, expansion is always automatic, and also works in \texttt{DVI} mode (\texttt{dvilualatex}), however, because postprocessing programs like \texttt{dvips} or \texttt{dvipdfmx} are not (yet) capable of dealing with OpenType fonts, only for legacy fonts.

\begin{itemize}
  \item \texttt{stretch} \texttt{\langle integer\rangle} \hspace{1cm} \texttt{20}
  \item \texttt{shrink} \hspace{1cm} 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 \texttt{1000}, so that a stretch limit of \texttt{10} means that the font may be expanded by up to \texttt{1\%}. The default stretch limit is \texttt{20}. The shrink limit will by default be the same as the stretch limit.
  \item \texttt{step} \texttt{\langle integer\rangle} \hspace{1cm} \texttt{* 1}
    Fonts are not expanded by arbitrary amounts but only by certain discrete steps within the expansion limits. With recent versions of \texttt{pdfTeX} (1.40 or newer) or \texttt{LuaTeX}, this option is by default set to \texttt{1}, in order to allow trying the maximum number of font instances, and hence to guarantee the best possible output.\footnote{Older \texttt{pdfTeX} versions, however, had to include every font instance in the PDF file, which may increase the file size quite dramatically. Therefore, in case you are using a pre-1.40 \texttt{pdfTeX} version, \texttt{step} is by default set to one fifth of the smaller value of \texttt{stretch} and \texttt{shrink}.}
  \item \texttt{selected} \texttt{true, false} \hspace{1cm} \texttt{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 \textit{selected expansion}, and its usage allows increasing the stretch and shrink limits (to, say, \texttt{30} instead of \texttt{20}); however, the gain is limited since at the same time the average stretch variance will be decreased. Therefore, this option is by default set to \texttt{false}, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.
\end{itemize}

### 3.4 Tracking

\texttt{letterspace} \texttt{\langle integer\rangle} \hspace{1cm} \texttt{100}

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

### 3.5 Miscellaneous options

\texttt{DVIoutput} \texttt{true, false} \hspace{1cm} \texttt{false}

\texttt{pdfTeX} and \texttt{LuaTeX} are not only able to generate PDF output but can also spit out \texttt{DVI} files.\footnote{Recent \TeX systems are using \texttt{pdfTeX} as the default engine even for \texttt{DVI} output.} The latter can be ordered with the option \texttt{DVIoutput}, which will set \texttt{\pdfoutput} to zero. For \texttt{XeTeX}, this option is not applicable.

---

\footnote{The downside with this default is that \texttt{pdfTeX} may run out of memory with huge documents; in this case, read about the error messages in the ‘Hints and caveats’ section (9), or try with a larger \texttt{step}.}
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. With `pdflTeX`, neither letterspacing nor automatic font expansion will work because the postprocessing drivers (`dvips`, `dvipdfm`, etc.) resp. the DVI viewer are not able to generate the fonts on the fly.

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

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

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

### Changing options later

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

```latex
\microtypesetup{expansion=false}
```
4 Selecting fonts for micro-typography

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

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

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

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

If you now call

\begin{verbatim}
\UseMicrotypeSet[protrusion]{basictext}
\end{verbatim}

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

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

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

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

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

\footnote{These translations will take place \texttt{\AtBeginDocument}, which means that changes to the defaults inside the preamble will also be taken into account. Only in cases where you change font defaults \texttt{\AtBeginDocument} yourself, you need to load \texttt{microtype} after these changes.}
Table 2: Predefined font sets

<table>
<thead>
<tr>
<th>Set name</th>
<th>Font attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Encoding: ∅, Family: ∅, Series: ∅, Shape: ∅, Size: ∅</td>
</tr>
<tr>
<td>alltext (allmath)</td>
<td>Text encodings, TS1 (OML, OMS, U)</td>
</tr>
<tr>
<td>alltext-nott (allmath-nott)</td>
<td>Text encodings, TS1 (OML, OMS, U)</td>
</tr>
<tr>
<td>basictext (basicmath)</td>
<td>Text encodings (OML, OMS) \rm*, \sf*</td>
</tr>
<tr>
<td>smallcaps</td>
<td>Text encodings ∅ ∅ \sc*, si, scit ∅</td>
</tr>
<tr>
<td>footnotesize</td>
<td>Text encodings, TS1 ∅ ∅ ∅ \footnotesize</td>
</tr>
<tr>
<td>scriptsize</td>
<td>Text encodings, TS1 ∅ ∅ ∅ \footnotesize</td>
</tr>
<tr>
<td>normalfont</td>
<td>\encoding*, \family*, \series*, \shape*, \normalsize</td>
</tr>
</tbody>
</table>

‘Text encodings’ = OT1, T1, T2A, LY1, OT4, QX, T5, EU1, EU2, TU ‘\ldots*’ = ‘\ldots default’

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

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

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

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

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

\usepackage[protrusion=allmath,tracking=smallcaps]{microtype}
\texttt{\UseMicrotypeSet}\{\{\texttt{features}\}\{\\{\texttt{set name}\}\}\}

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

\texttt{\DeclareMicrotypeSetDefault}\{\{\texttt{features}\}\{\\{\texttt{set name}\}\}\}

If a feature is enabled but no font set has been chosen explicitly, the sets declared by this command will be activated. By default, the ‘all\texttt{text}’ font set will be used for character protrusion and additional kerning, the ‘basictext’ set for font expansion and interword spacing, and the ‘smallcaps’ set for tracking.

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

5 Micro fine tuning

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

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

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

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

\SetProtrusion \texttt{\{options\} \{set of fonts\} \{protrusion settings\}}

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

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

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

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

Options:

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

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

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

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

\SetProtrusion
\[
\texttt{\{factor = 700,}
\texttt{load = cmr-T1\}}
\{
  encoding = T1,
  family = cmr,
  size = large\}
\]
MICRO FINE TUNING: Font expansion

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`.

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

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

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

5.2 Font expansion

\SetExpansion\{\{options\}\}\{\{set of fonts\}\}\{\{expansion settings\}\}

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

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

Options:

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

The unit option can even be passed globally to the package (cf. section 3.2). However, all provided settings are created under the assumption that the values are relative to the character width. Therefore, you should only change it if you are certain that the default settings will not be used in your document.
all expansion factors, to set the input encoding, or to determine the context of the list (expansion contexts are only possible with pdf\TeX{} version 1.40.4 or newer).

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

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

\begin{verbatim}
\SetExpansion
  [ context = sloppy,
    stretch = 30,
    shrink = 60,
    step   = 5  ]
  { encoding = \{OT1,T1,TS1\} }
{ }
\end{verbatim}

This method of employing contexts to temporarily apply different expansion parameters only works with pdf\TeX{} version 1.40.4 or later.\footnote{For older versions, a dirty trick is laid out in section 14.2 on page 58.} Also note that pdf\TeX{} prohibits the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the sloppy context has to be applied to complete paragraphs.

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

\begin{verbatim}
\SetExpansion
  [ factor = 500 ]
  { encoding = *,
    shape = it }
{ }
\end{verbatim}

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 \texttt{stretch} and \texttt{shrink} options.

5.3 Tracking

\SetTracking \{\texttt{options}\} \{\texttt{set of fonts}\} \{\texttt{tracking amount}\}

An important typographic technique – which was missing in \TeX{} for a long time – is the adjustment of tracking, i.e., the uniform addition or subtraction of letter space
to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.\footnote{With full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.} The \SetTracking command allows specifying the tracking amount for different fonts or font sets. It will also be evaluated by the \textls command, which may be used for letterspacing shorter pieces of text (see section 7).

The tracking amount is specified in thousandths of 1em (or the given unit); negative values are allowed, too.

Options:

- \texttt{name, unit, context} These options serve the same functions as in the previous configuration commands. The unit may be any dimension, default is 1em.
- \texttt{spacing} When the inter-letter spacing is altered, the inter-word spacing probably also needs to be adjusted. This option expects three numbers for interword space, stretch and shrink respectively, which are given in thousandths of 1em (or of the current unit). If a value is followed by an asterisk, it denotes thousandths of the respective font dimension which will be added to it. For instance, with

\begin{verbatim}
\SetTracking[ spacing = {25*,166, } ]{ encoding = *, shape = sc }{ 25 }
\end{verbatim}

the interword space will be increased by 2.5%, the stretch amount will be set to 0.166em, while the shrink amount will be left untouched. If you don’t specify the \texttt{spacing} option, the interword space will be scaled by the current letterspace amount (as in the above example), while stretch and shrink will not be changed.

- \texttt{outer spacing} If an interword space immediately precedes or follows letter-spaced text, it will by default be equal to that within the text. With this option, which accepts the same values as \texttt{spacing}, it may be adjusted independently.

- \texttt{outer kerning} If, on the other hand, no interword space precedes or follows, you may still want to slightly set off the first and last letter from adjoining letters. This option expects the kerning amounts for left and right hand side, separated by a comma, in thousandths of 1em (or the current unit). If a value is followed by an asterisk, it denotes thousandths of the current letterspacing amount. A single asterisk means ‘500*’; this is also the default, i.e., the sum of the outer kerns is by default equal to the current letterspace amount. To remove kerning on both sides, you would write ‘outer kerning={0,0}’.

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

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

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

and then write:

Stop \textls{stealing sheep}!

this would be the (typographically dubious) outcome:

\begin{center}
Stop \textls{stealing sheep}!
\end{center}

While the word ‘Stop’ is not letterspaced, the space between the letters in the other two words is expanded by the tracking amount of 160/1000em = 0.16em. The inner space within the letterspaced text is increased by 60%, while its stretch amount is decreased by 10% and the shrink amount is left untouched. The outer space (of 0.45em) immediately before the piece of text may stretch by 0.25em and shrink by 0.15em. Note that there is no outer space after the text, since the exclamation mark immediately follows; instead, the default outer kern of half the letterspace amount (0.08em) is added. Furthermore, one ligature wasn’t broken up, because we neglected to specify the ‘s’ in the no ligatures key.

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

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

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

8 With pdfTeX versions older than 1.40.4, all ligatures, and hence all kerning, will be disabled. It is therefore recommended to use at least version 1.40.4.
your editor wants you to shorten your 1000-pages chef-d'œuvre by a handful of pages, you could load microtype with (fingers crossed):
\usepackage[tracking=alltext,letterspace=-40]{microtype}

5.4 Additional kerning
\SetExtraKerning \[options\] \{set of fonts\} \{kerning settings\}

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

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

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

Options:
name, load, factor, preset, inputenc These options serve the same function as in the previous configuration commands.
unit Admissible values are: space, character and a \(\langle \text{dimension} \rangle\). By default, the values denote thousandths of 1em.
context When it comes to kerning settings, this option is especially useful, since it allows applying settings depending on the current language.

For example, you can find the following settings, intended to be used for documents written in French, in the main configuration file:
\SetExtraKerning
\[ name = \text{french-default}, \]
\[ context = \text{french}, \]
\[ unit = \text{space} \]
\[ \{ encoding = \{\text{OT1},\text{T1},\text{LY1}\} \} \]
\[ \{ : = \{1000,\}, % = \text{\fontdimen2} \]
\[ ; = \{500,\}, % = \text{\thinspace} \]
\[ ! = \{500,\}, \]
\[ ? = \{500,\} \]

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

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

5.5 Interword spacing

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

On top of these glue dimensions, \TeX has the concept of ‘space factors’. They may be used to increase the space after certain characters, most prominently the punctuation characters. \pdfTeX’s additional spacing adjustment may be considered as an extension to space factors with much finer control: while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – you may modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. Note that when interword spacing adjustment is in effect, space factors are ignored.

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

\textit{Options:}

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

unit You can specify the unit by which the specified numbers are measured. Possible values are: character, a (dimension) and, additionally, space. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with the following (nonsensical) settings:
the space inserted after a full stop would be doubled (technically speaking: $2 \times \text{\fontdimen2}$), as would the maximum stretch and shrink amounts of the interword space ($\text{\fontdimen3}$ and $\text{\fontdimen4}$). Conversely, setting all three values to $-1000$ would completely cancel a space after the respective character.

\section*{5.6 Character inheritance}

\DeclareCharacterInheritance

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters À, Á, Â, Ã, Ä, Å and ˘A should probably be protruded by the same (absolute) amount as the character A. Using the command \DeclareCharacterInheritance, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list of features, you can confine the scope of the list. Additionally, it accepts the inputenc key to set the input encoding for this list. The font set can be declared in the usual way. The inheritance lists are declared as pairs of (base character) = (list of inheriting characters). Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

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

\section*{5.7 Configuration files}

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

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

\DeclareMicrotypeVariants
\DeclareMicrotypeVariants*

On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a 'variant' of the base font (cf. Karl Berry's \texttt{Fontname}). It is thus possible to put settings for, e.g., the
Table 3: Fonts with tailored protrusion settings

<table>
<thead>
<tr>
<th>Font family (NFSS code)</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encodings [Scripts]</td>
</tr>
<tr>
<td>Generic</td>
<td>OT1, T1, T2A, LY1, QX, (TS1)$^a$</td>
</tr>
<tr>
<td>Computer Modern Roman (cmr)$^b$</td>
<td>OT1, OT4, T1, T2A, T5, LY1, TS1</td>
</tr>
<tr>
<td>Bitstream Charter (bch)$^c$</td>
<td>OT1, T1, T5, LY1, TS1</td>
</tr>
<tr>
<td>Adobe Garamond (pad, padx, padj)</td>
<td>OT1, T1, LY1, TS1</td>
</tr>
<tr>
<td>URW Garamond (ugm)$^e$</td>
<td>OT1, T1, TS1</td>
</tr>
<tr>
<td>Bitstream Letter Gothic (bly)$^f$</td>
<td>OT1, T1, TS1</td>
</tr>
<tr>
<td>Adobe Minion (pmmx, pmmj)</td>
<td>OT1, T1, T2A, LY1, TS1</td>
</tr>
<tr>
<td>Palatino (ppl, pplx, pplj)$^g$</td>
<td>OT1, OT4, T1, LY1, (TS1)$^a$</td>
</tr>
<tr>
<td>Times (ptm, ptmx, ptmj)$^h$</td>
<td>OT1, OT4, T1, LY1, QX, (TS1)$^a$</td>
</tr>
<tr>
<td>Latin Modern Roman</td>
<td>EU1/2, TU [Latin, Greek]</td>
</tr>
<tr>
<td>Charis SIL</td>
<td>EU1/2, TU [Latin, Cyrillic, Greek]</td>
</tr>
<tr>
<td>Palatino Linotype$^i$</td>
<td>EU1/2, TU [Latin]</td>
</tr>
<tr>
<td>Computer Modern math (cmsy, cmm)$^j$</td>
<td>OML/OMS</td>
</tr>
<tr>
<td>AMS symbols (msa, msb)</td>
<td>U</td>
</tr>
<tr>
<td>Euler (eur, eus, euf)$^k$</td>
<td>U</td>
</tr>
<tr>
<td>Euro symbols (Adobe, ITC, marvosym)</td>
<td>U/OT1</td>
</tr>
</tbody>
</table>

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

fonts padx (expert set), padj (oldstyle numerals) and pad (plain) into one and the same file mt-pad.cfg. This command expects a comma-separated list of variant suffixes. The starred version appends the suffix(es) to the existing list. The default declaration in microtype.cfg is:

```latex
\DeclareMicrotypeAlias{}{(font name)}{(alias font)}
```

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

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

which would make the package, whenever it encounters the font lmr and does not find settings for it, also try the font cmr. In fact, you will find this very line, along with some others, in the default configuration file.
\LoadMicrotypeFile \{\texttt{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.\footnote{Font package authors might also want to have a look at the hook \texttt{Microtype@Hook}, described in the
implementation part, section 14.4.4.} This command will load the file \texttt{'mt-\{font name\}.cfg'}.

6 Context-sensitive setup

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

\microtypecontext \{\texttt{context assignments}\}

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

\begin{microtypecontext} \{\texttt{context assignments}\} \end{microtypecontext}

Like many \LaTeX{} commands, it is also available in the form of an environment.

\textmicrotypecontext \{\texttt{context assignments}\} \{\texttt{general text}\}

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

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

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

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

\begin{lstlisting}[numbers=left]
\newcommand*\new@makefnmark{\hbox{\@textsuperscript{\normalfont\microtypecontext{protrusion=footnote}@thefnmark})}}
\renewcommand*\@footnotemark{\% \leavevmode \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
\new@makefnmark \ifhmode\spacefactor\@x@sf\fi \relax}
\end{lstlisting}

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

\begin{lstlisting}[numbers=left]
\renewcommand*\@makefnmark{\hbox{\@textsuperscript{\normalfont\microtypecontext{protrusion=footnote}@thefnmark}}}
\let\m@mmf@prepare\relax
\let\m@mmf@check\relax
\end{lstlisting}
Another possibility would be to employ contexts for a language-dependent setup. For instance, if you are writing a text in French, you could add:

```latex
\microtypecontext{kerneling=french}
```

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

```latex
\textmicrotypecontext{kerneing=}{English text!}
```

to reset the context, so that the punctuation characters in these parts will not receive any extra kerning.

Instead of adding these commands manually to your document, you may also load `microtype` with the `babel` option (see section 3.5). The current language will then be automatically detected and the contexts set accordingly.

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

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

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

## 7 Letterspacing revisited

While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.\footnote{Letterspacing should be used cautiously; in particular, letterspacing lowercase text is held in abhorrence by honourable typographers. Unless you know what you are doing, you should probably only letterspace capitals or small capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.} For such ad-hoc letterspacing, `microtype` introduces two commands that can be used (independently of whether the tracking option is enabled) in the same way as \LaTeX’s text commands: \texttt{\textls} – which also works in math mode – expects the text in the mandatory argument, while \texttt{\lsstyle} will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of \texttt{\textls} does not add any extra kerning before or after the text, which may be useful, e.g., for section titles. By default, each character will be spaced out by 100/1000em = 0.1em; this amount may be altered in the optional argument to \texttt{\textls}, using the \texttt{\SetTracking} command, or globally with the \texttt{letterspace} package option, with decreasing significance in this order.
\lslig \{\textls\textstyle\}

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

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

\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{120}
\textfrak\lsstyle Au\lslig{\}s\lslig{\}s\lslig{\}i\lslig{\}cht\lslig{\}i\lslig{\}gkeit}

\letterspace.sty

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

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

8 Disabling ligatures

\DisableLigatures \{\texttt{(characters)}\} \{\texttt{(set of fonts)}\}

While completely disabling all ligatures of a font (which will also switch off kerning for this font), purposely lowers the micro-typographic quality instead of raising it, it is especially useful for typewriter fonts, so that, e.g., in a T1 encoded font, ‘\texttt{--}’ will indeed be printed as ‘\texttt{--}’, not as ‘\texttt{-.}’. \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*]
HINTS AND CAVEATS

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

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

Only the character that begins the ligature(s) should be specified. This command may only be used in the preamble, and only once.\footnote{With \LaTeX, you have to load the fonts with the \texttt{fontspec} option \texttt{Renderer=Basic}.}

9 Hints and caveats

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

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

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

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

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

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

\begin{verbatim}
\AtBeginEnvironment{verbatim}{\microtypesetup{activate=false}}
\end{verbatim}

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

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

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

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

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

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

• Before this package was fully compatible with LuaTEX, the following method of enabling expansion and protrusion with the fontsize package was most often found to be recommended:

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

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

• With pdfTeX, it is currently not possible to create character-specific settings for Chinese/Japanese/Korean fonts. Therefore, the only micro-typographic extension that can be made to work with CJK fonts is (non-selected) font expansion.

• When used with the xeCJK package or the luatexja package, text commands (e.g., \textless) in the configuration will not be understood. You therefore have to ensure that microtype will encounter none of them. This requires, firstly, that the glyphs be specified only as single (possibly Unicode) characters, as numbers, or as glyph names (cf. section 5); and secondly, if you are using a font for which pre-defined settings do not exist, that you create these settings yourself (because otherwise, the default settings will be loaded, which do contain text commands). Furthermore, you should load microtype late.

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

* ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.
  This error message will occur if you are trying to employ font expansion while creating DVI output. Remember that automatic font expansion only works when running pdfTeX 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.

* ! pdftex error (font expansion): auto expansion is only possible with scalable fonts.
  Automatic font expansion has been improved in pdfTeX 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn’t found at all, e.g., because of missing map entries.

* Warning: pdflatex: font ptmbr8r cannot be expanded (not an included Type1 font) and the PDF viewer complains about a missing font, e.g., Adobe Reader thusly:
  Could not find a font in the Resources dictionary - using Helvetica instead.
  With pdfTeX versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your TeX system is not set up to embed (or ‘download’) the base PostScript fonts (e.g., Times, Helvetica, Courier). In most TeX distributions, this can be changed in the file updmap.cfg by setting pdftexDownloadBase14 to true.

¹² They make use of features provided by luatotfload (via fontsinc).
Furthermore, pdfTeX versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdfTeX versions, this is only possible if you manually create expanded instances of the fonts.

* ! Font `T1/cmr/m/n/10=ecrm1000' at 10.0pt not loaded: Not enough room left.
  Memory parameter ‘font_mem_size’ too small.

* ! TeX capacity exceeded, sorry [maximum internal font number (font_max)=2000].
  Memory parameter ‘font_max’ too small.

* ! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].
  Memory parameter ‘pdf_mem_size’ too small (pdfTeX versions older than 1.30).

When applying micro-typographic enhancement to a large document with a lot of fonts, pdfTeX may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2C-based systems, e.g., TeX Live, change the settings in texmf.cnf, for MiKTeX, in the file miktex.ini (2.4 or older) resp. pdflatex.ini (2.5 or newer).

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

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

10 Contributions

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

11 Acknowledgments

This package would be pointless if Hán Thế Thành hadn’t created the pdfTeX programme in the first place, which introduced the micro-typographic extensions and made them available to the TeX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in Thành 2004, Thành 2008 and elsewhere. I also thank him and the rest of the pdfTeX team, and more recently also the LuaTeX and XeTeX teams, for refuting the idea that TeX is dead, and for fixing the bugs I find.

Harald Harders has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make.
Andreas Bühmann has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. Ulrich Dirr has made numerous suggestions, especially concerning the new extensions of interword spacing adjustment and additional character kerning. Georg Duffner has patiently tested microtype under Xe\TeX{} and Lua\TeX{} with his beautiful OpenType font EB Garamond\footnote{Available from CTAN at \texttt{pkg/ebgaramond}, including configuration files for \texttt{microtype}.}. My thanks also go to Maciej Eder for contributing settings for the QX encoding, as well as to Karl Karlsson for providing settings for the Cyrillic T2A encoding, and to Hendrik Vogt, who made substantial improvements to the Computer Modern Roman italic settings. I thank Loren B. Davis for providing protrusion settings for OpenType versions of Palatino Linotype. I am also very much indebted to Élie Roux, who not only contributed the \texttt{lua} module in the first place, but also, together with Philipp Gesang, took care of updating it for the developments in Lua\TeX{} land.

I thank Philipp Lehman for adding to his \texttt{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 \texttt{ledmac/ledpar} packages, so that critical editions can finally also benefit from character protrusion. Likewise, Donald Arseneau patched his \texttt{shapepar} package to accommodate protrusion.


12 References


13 Short history

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

2.7 (2017/07/07)
• Allow automatic expansion and letterspacing with LuaTeX in DVI mode (aka. dvilualatex) [3.1, 3.3, table 1]
• Compatibility with \LaTeX 2017/01/01 (fix warnings)

2.6 (2016/05/01)
• Support for LuaTeX $\geq 0.85$
• Improvements for tracking/letterspacing with LuaTeX (Renderer=Basic no longer required)
• New font sets: ‘alltext-nott’, ‘allmath-nott’ [4, table 2]

2.5 (2013/03/13)
• Support for the fontspec package, viz. for OpenType fonts with LuaTeX and XeTeX
• Support for protrusion with XeTeX $\geq 0.9997$
• Support for tracking/letterspacing with LuaTeX $\geq 0.62$
• Allow context-sensitive setup with LuaTeX
• Info if protrusion settings are generic
• Protrusion settings for Latin Modern Roman (OpenType)
• Protrusion settings for Charis SIL (OpenType)
• Protrusion settings for Palatino Linotype (OpenType)
2.4 (2010/01/10)
• Protrusion settings for T2A encoded Minion

2.3e (2009/11/09)
• Support for the Cyrillic T2A encoding (protrusion, expansion, spacing)

2.3d (2009/03/27)
• New default for expansion option ‘step’: 1, if pdftex ≥ 1.40 [3.3]

2.3c (2008/11/11)
• Support for LuaTeX enabled by default

2.3 (2007/12/23)
• New key ‘outer kerning’ for \SetTracking to customise outer kerning [5.3]
• Adjust protrusion settings for tracking even if protrusion is not enabled
• New option ‘verbose=silent’ to turn all warnings into mere messages [3.5]
• The letterspace package also works with eplain or miniltx [7]

2.2 (2007/07/14)
• Improvements to tracking/letterspacing: retain kerning (pdftex ≥ 1.40.4); automatically adjust protrusion settings
• New key ‘no ligatures’ for \SetTracking to disable selected or all ligatures (pdftex ≥ 1.40.4) [5.3]
• New keys ‘spacing’ and ‘outer spacing’ for \SetTracking to customise interword spacing [5.3]
• Possibility to expand a font with different parameters (pdftex ≥ 1.40.4) [5.2]
• New optional argument for \DisableLigatures to disable selected ligatures [8]
• New command \DeclareMicrotypeVariants to specify variant suffixes [5.7]
• New command \textmicrotypecontext as a wrapper for \microtypecontext [6]
• Protrusion settings for Bitstream Letter Gothic

2.1 (2007/01/21)
• New command \lslig to protect ligatures in letterspaced text [7]

2.0 (2007/01/14)
• Support for the new extensions of pdftex ≥ 1.40: tracking/letterspacing, additional kerning, and adjustment of interword spacing (glue) (new commands \SetTracking, \SetExtraKerning, \SetExtraSpacing; new options ‘tracking’, ‘kerning’, ‘spacing’) [5.3, 5.4, 5.5]
• New commands \textls and \lsstyle for letterspacing, new option ‘letterspace’ [3.4, 7]
• New option ‘babel’ for automatic micro-typographic adjustment to the selected language [3.5, 6]
• New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ [4, table 2]
• New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing [7]

1.9e (2006/07/28)
• New key ‘inputenc’ to specify the lists’ input encodings [5]
• Protrusion settings for Euler math fonts
1.9d (2006/05/05)
- Support for the Central European QX encoding (protrusion, inheritance)
- Protrusion settings for various Euro symbol fonts (Adobe, ITC, marvosym)
- Support for Unicode input in the configuration (inputenc/utf8)

1.9c (2006/02/02)
- Protrusion settings for URW Garamond

1.9a (2005/12/05)
- Defer setup until the end of the preamble
- Inside the preamble, \microtypesetup accepts all package options \[3.6\]
- Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)
- New command \DisableLigatures to disable ligatures (pdf\TeX \geq 1.30) \[8\]
- New command \microtypecontext to change the configuration context; new key ‘context’ for the configuration commands \[6\]
- New key ‘font’ to add single fonts to the font sets \[4\]
- New key ‘preset’ to set all characters to the specified value before loading the lists
- Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
- Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
- Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)

1.8 (2005/06/23)
- New command \DeclareMicrotypeSetDefault to declare the default font sets \[4\]
- New option ‘config’ to load a different configuration file \[3.5\]
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width \[5.1\]
- Renamed commands from \..MicroType.. to \..Microtype..
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The ‘allmath’ font set also includes U encoding
- Support for protrusion with the ledmac package (pdf\TeX \geq 1.30)

1.7 (2005/03/23)
- Possibility to specify ranges of font sizes in the set declarations \[4, 5\]
- New command \LoadMicrotypeFile to load a configuration file manually \[5.7\]
- New command \Microtype@Hook for font package authors \[14.4.4\]
- New option ‘verbose=errors’ to turn all warnings into errors
- Warning when running in draft mode

1.6 (2005/01/24)
- New option ‘factor’ to influence protrusion resp. expansion of all characters of a font or font set \[3.2, 5\]
- When pdf\TeX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled \[3.1\]
- Use e-\TeX extensions, if available

1.5 (2004/12/15)
- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled \[3.1\]
• New option 'selected' to enable selected expansion, default: false [3.3, 5.2]
• New default for expansion option 'step': 4 (min(stretch, shrink)/5) [3.3]
• Protrusion settings for Bitstream Charter

1.4 (2004/11/12)
  • Set up fonts independently from \LaTeX{} font loading
  • New option: 'final' [3.5]

1.2 (2004/10/03)
  • New font sets: 'allmath' and 'basicmath' [4, table 2]
  • Protrusion settings for Computer Modern Roman math symbols
  • Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond

1.1 (2004/09/21)
  • Protrusion settings for Adobe Minion
  • New command: \DeclareCharacterInheritance [5.6]
  • Characters may also be specified as octal or hexadecimal numbers [5]

1.0 (2004/09/11)
  • First CTAN release
14 Implementation

The docstrip modules in this file are:

driver: The documentation driver, only visible in the dtx file.

package: The code for the microtype package (microtype.sty).
pdftex-def: Definitions specific to pdf\TeX\ (microtype-pdftex.def).
xetex-def: Definitions specific to \LaTeX\ (microtype-xetex.def).
luatex-def: Definitions specific to Lua\TeX\ (microtype-luatex.def).

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

plain: Code for \plain, \miniltx (letterspace only).
luafile: Lua functions (microtype.lua).

config: Surrounds all configuration modules.

\cfg-t: Surrounds (Latin) text configurations.
  \mt: The main configuration file (microtype.cfg).
  \bch: Settings for Bitstream Charter (mt-bch.cfg).
  \blg: Settings for Bitstream Letter Gothic (mt-blg.cfg).
  \cmr: Settings for Computer Modern Roman (mt-cmr.cfg).
  \pad: Settings for Adobe Garamond (mt-pad.cfg).
  \ppl: Settings for Palatino (mt-ppl.cfg).
  \ptm: Settings for Times (mt-ptm.cfg).
  \pmn: Settings for Adobe Minion (mt-pmn.cfg).
  Contributed by Harald Harders.
  \ugm: Settings for URW Garamond (mt-ugm.cfg).

\cfg-u: Surrounds non-text configurations (U encoding).
  \msa: Settings for AMS 'a' symbol font (mt-msa.cfg).
  \msb: Settings for AMS 'b' symbol font (mt-msb.cfg).
  \euf: Settings for Euler Fraktur font (mt-euf.cfg).
  \eur: Settings for Euler Roman font (mt-eur.cfg).
  \eus: Settings for Euler Script font (mt-eus.cfg).

\cfg-e: Surrounds Euro symbol configurations.
  \zpeu: Settings for Adobe Euro symbol fonts (mt-zpeu.cfg).
  \euroitc: Settings for ITC Euro symbol fonts (mt-euroitc.cfg).
  \mvs: Settings for marvosym Euro symbol (mt-mvs.cfg).

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

And now for something completely different.
14.1 Preliminaries

This is us.

\MT@MT

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

\MT@restore@catcodes

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

\MT@fix@catcode

\MT@fix@catcode{17}{14}% ^^Q (comment)
\MT@fix@catcode{24} {9}% \^^X (ignore)
\MT@fix@catcode{33}{12}% !
\MT@fix@catcode{34}{12}% ”
\MT@fix@catcode{36} {3}% $ (math shift)
\MT@fix@catcode{39}{12}% ’
\MT@fix@catcode{42}{12}% *
\MT@fix@catcode{43}{12}% +
\MT@fix@catcode{44}{12}% ,
\MT@fix@catcode{45}{12}% -
\MT@fix@catcode{58}{12}% :
\MT@fix@catcode{60}{12}% <
\MT@fix@catcode{61}{12}% =
\MT@fix@catcode{62}{12}% >
\MT@fix@catcode{63}{12}% ?
\MT@fix@catcode{94} {7}% ^ (superscript)
\MT@fix@catcode{96}{12}%`
\MT@fix@catcode{124}{12}% |

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

\newcommand\DeclareMicrotypeSet[3][]{()} 
\newcommand\UseMicrotypeSet[2][]{()) 
\newcommand\DeclareMicrotypeSetDefault[2][]{()) 
\newcommand\SetProtrusion[3][]{()} 
\newcommand\SetExpansion[3][]{()} 
\newcommand\SetTracking[3][]{()} 
\newcommand\SetExtraKerning[3][]{()} 
\newcommand\SetExtraSpacing[3][]{()} 
\newcommand\DisableLigatures[2][]{()) 
\newcommand\DeclareCharacterInheritance[3][]{()) 
\newcommand\DeclareMicrotypeVariants[1][]{()) 
\newcommand\DeclareMicrotypeAlias[2][]{()) 
\newcommand\LoadMicrotypeFile[1][]{()) 
\newcommand\DeclareMicrotypeBabelHook[2][]{()) 
\newcommand\microtypesetup[1][]{()) 
\newcommand\microtypecontext[1][]{()) 
\newcommand\textmicrotypecontext[2][]{#2} 
\ifpackagefileloaded[letterspace]\\let\MT@textls\relax\% 
\fi 
\package{package} 
\nsstyle{} 
\textls{2}[]
These commands also have a starred version.

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

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

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

Don't load `letterspace`.

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

The old command names had one more hunch.

```
\MT@old@cmd
```

Communicate.

```
\MT@warning\MT@dinfo
\MT@dinfo@nl
```

Cases for `\tracingmicrotype`:

```
\tracingmicrotype
```

0: almost none
1: + sets & lists
2: + heirs
3: + slots
4: + factors

```
\@onlydebug\\MT@dinfo\\MT@dinfo@nl
```

This error message appears because you loaded the `\MT@MT` package with the option `verbose=errors`. Consult the documentation in `\MT@MT.pdf` to find out what went wrong.)

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

14.1.1 Debugging

```
\tracingmicrotype
\MT@info
\MT@info@nl
```

4
Another debug method: font switches can be marked in the PDF file with a small caret, an accompanying popup text box displaying all debug messages.

Cases for `\tracingmicrotypeinpdf`:

1. Show new fonts
2. + Show known fonts

If `microtype.sty` had been generated with the `debug` option, this method would be demonstrated here.

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

With `\tracingmicrotypeinpdffallfalse`, the PDF output is (hopefully) identical, but some font switches will not be displayed; otherwise the output is affected, but all font switches are visible. In the latter case, we also insert a small kern so that multiple font switches are discernable.

A red caret is shown for fonts which are actually set up by Microtype, a green one marks fonts that we have already seen. The `/Caret` annotation requires a viewer for `PDF` version 1.5 (you could use `/Text` if you’re using an older `PDF` viewer).

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

98 \newcount\tracingmicrotypeinpdf
99 \RequirePackage{pdftexcmds}
100 \newif\ifMT@inannot \MT@inannottrue
101 \let\MT@pdf@annot\@empty
102 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
103 {\def\MessageBreak{\@spaces}\MT@xadd\MT@pdf@annot{\pdf@escapestring{#1\@spaces}}\fi\fi}
104 \iftracingmicrotypeinpdfall
105 \ifnum\tracingmicrotypeinpdf<#1 \else
106 \iftracingmicrotypeinpdfall \leavevmode\fi
107 \pdfannot height 4pt width 4pt depth 2pt \%
108 {/Subtype/Caret (/\expandafter\string\font@name) \ifcase#1\or
109 /Subj(New font)/C[1 0 0]\else
110 /Subj(Known font)/C[0 1 0]\fi
111 /Contents({\MT@pdf@annot})\%
112 }\fi
113 \global\MT@inannotfalse
114 \fi
115 \fi
116 \fi
117 \fi
118 \fi
119 \fi
120 \fi
121 \fi
122 (debug)
123 (package)
```
14.1.2 Requirements

The \texttt{letterspace} package works with:

0: \texttt{miniltx}

1: \texttt{eplain}

2: \LaTeX

For plain usage, we have to copy some commands from \texttt{latex.ltx}.

\begin{verbatim}
\MT@plain
\def\MT@plain{2}
\ifx\documentclass\@undefined
\def\MT@plain{1}
\def\hmode@bgroup{\leavevmode\bgroup}
\let\@typeset@protect\relax
\ifx\eplain\@undefined
\def\MT@plain{0}
\def\PackageWarning#1#2{\begingroup\newlinechar=10 %\def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces} %\immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}\endgroup\}
\def\on@line{ on input line \the\inputlineno}
\def\@spaces{\space\space\space\space}
\fi
\fi
\MT@requires@latex
\better use groups than plain ifs.
\MT@clear@options
\end{verbatim}

Better use groups than plain ifs.

\begin{verbatim}
\ifnum\MT@plain=1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
\MT@endinput
\end{verbatim}

For definitions that depend on e-\TeX features.

\begin{verbatim}
\ifcase 0\fi
\ifx\eTeXversion\@undefined \else\fi
\ifx\eTeXversion\relax \else\fi
\MT@clear@options
\end{verbatim}

Better use groups than plain ifs.

We check whether we are running \texttt{pdf\LaTeX}, \texttt{X\LaTeX}, or \texttt{Lua\LaTeX}, and load the appropriate definition file.
has been fixed in TeX Live 2005.
\begin{verbatim}
\ifx\normalpdftexversion\@undefined \else
\let\pdftexversion \normalpdftexversion
\let\pdftexrevision \normalpdftexrevision
\let\pdfoutput \normalpdfoutput
\fi
\MT@engine\MT@engine@tooold
\if\pdftexversion\@undefined \else
\if\pdftexversion\relax \else
\def\MT@engine{pdf}
\fi\fi
\fi
\if\directlua\@undefined \else
\if\directlua\relax \else
\def\MT@engine{lua}
\fi\fi
\fi
\ifx\pdftexversion\@undefined \else
\ifx\pdftexversion\relax \else
\def\MT@engine{pdf}
\fi\fi
\fi
\ifx\directlua\@undefined \else
\if\directlua\relax \else
\def\MT@engine{lua}
\fi\fi
\fi
\MT@pdftex@no pdf\TeX's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdf\TeX we're using, if any. \MT@pdftex@no will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for pdf\TeX:

0: not running pdf\TeX
1: pdf\TeX ($< 0.14f$)
2: + micro-typographic extensions ($0.14f,g$)
3: + protrusion relative to 1em ($\geq 0.14h$)
4: + automatic font expansion; protrusion no longer has to be set up first; scale factor fixed to 1000; default $\efcode = 1000$ ($\geq 1.20$)
5: + \( \left( \text{left, right} \right) \text{marginkern; pdfnoligatures; pdfstrcmp; pdfescapestring} \) 
\( \geq 1.30 \)

6: + adjustment of interword spacing; extra kerning; \letterspacefont; pdfmatch\(^{14}\); \pdftracingfonts; always \( \text{e-TEX} \) \( \geq 1.40 \)

7: + \letterspacefont \( \text{doesn't disable ligatures and kerns; pdfcopyfont} \) \( \geq 1.40 \)

\(^{14}\) This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.
\MT@lua Communicate with \texttt{lua}. Beginning with \LaTeX\ 0.36, \texttt{directlua} no longer requires a state number.
\begin{Verbatim}
def\MT@lua{\directlua}
def\MT@luatex@no{5}
\ifnum\luatexversion<90
  \def\MT@luatex@no{4}
\ifnum\luatexversion<85
  \def\MT@luatex@no{3}
\ifnum\luatexversion<62
  \def\MT@luatex@no{2}
\ifnum\luatexversion<36
  \def\MT@lua{\directlua0}
  \def\MT@luatex@no{1}
\fi
\fi
\fi
\fi
\MT@dinfo@nl{0}{luatex no.: \MT@luatex@no}
\end{Verbatim}
\begin{Verbatim}
\ifnum\MT@engine\texttt{tex}@no < 2
  \MT@warning@nl{You are using a \MT@engine tex version older than \texttt{pdftex} or \texttt{luatex}. \MT@MT' does not work with this version. Please install a newer version of \MT@engine tex.}
\else
  \MT@clear@options
  \endinput\fi
\end{Verbatim}

Still there? Then we can begin: We need the \texttt{keyval} package, including the ‘new’ \texttt{KV@sp@def} implementation.
\begin{Verbatim}
\RequirePackage{keyval}[1997/11/10]
\end{Verbatim}

\MT@toks We need a token register.
\begin{Verbatim}
\newtoks\MT@toks
\ifMT@if@
  A scratch if.
\fi
\end{Verbatim}
14.1.3 Declarations

These are the global switches ...
For definitions that depend on a particular pdfTeX resp. LuaTeX version.

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

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

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

To be continued, but first back to primitives.
This is \namedef and global.

Its expanding versions.

\let a \csname sequence to a command.

\let a command to a \csname sequence.

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

Wrapper for testing whether command resp. \csname sequence is defined. If we are running e-T\TeX{}, we will use its primitives \ifdefined and \ifcsname, which decreases memory use substantially.
Translate a macro into a token list. With e-\TeX, we can use \texttt{\detokenize}. We also need to remove the last trailing space; and only the last one – therefore the fiddling (and the \texttt{string} isn't perfect, of course).

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

Test whether argument is empty.

\begin{verbatim}
\def\MT@ifempty#1{\if\%#1\%&\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
\end{verbatim}

Test whether argument is an integer, using an old trick by Mr. Arseneau, or the latest and greatest from pdf\TeX\ or Lua\TeX\ (which also allows negative numbers, as required by the \texttt{letterspace} option).

\begin{verbatim}
\def\MT@ifint#1{\ifcase\pdfmatch{^-*[0-9]+ *}$}{#1}\relax\expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi}
\def\MT@ifint#1{\if!\ifnum9<1#1!\else?\fi\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}\end{verbatim}
\texttt{\textbackslash MT@ifdimen} Test whether argument is dimension (or number). (nd and nc are new Didot resp. Cicero, added in pdf\TeX 1.30; px is a pixel.)

\begin{verbatim}
489 (pdftex-def)
\MT@要求@pdftex6{
  \def\MT@ifdimen#1{%
    \ifcase\pdfmatch{^([0-9]*([.,][0-9]+)?|[.,][0-9]+)%}(em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
    \expandafter\@secondoftwo
  }{\expandafter\@firstoftwo}
\}
490 (pdftex-def|xetex-def)
\def\MT@ifdimen#1{%\settobox\z@#1\relax
  \ifnum\MT@count=\@ne
    \aftergroup\@secondoftwo
  \else
    \aftergroup\@firstoftwo
  \fi}
491 (pdftex-def|xetex-def)
\def\MT@ifdimen#1#2#3{%\ifdim #1\p@ #2 #3\p@
  \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi}
\end{verbatim}

\texttt{\textbackslash MT@ifstreq} Test whether two strings (fully expanded) are equal.

\begin{verbatim}
local function if_streq(s, "^-*[0-9]+ +$") then
  tex_write("@firstoftwo")
else
  tex_write("@secondoftwo")
end
microtype.if_int = if_int
\end{verbatim}
\MT@ifstreq\#1\#2{\
  \ifcase\pdfstrcmp{\#1}{\#2}\relax
  \expandafter\@firstoftwo
  \else
  \expandafter\@secondoftwo
  \fi
}\}

〈pdftex-def〉
〈xetex-def〉
\def\MT@ifstreq#1#2{\
  \edef\MT@res@a{#1}\
  \edef\MT@res@b{#2}\
  \ifx\MT@res@a\MT@res@b
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
〈pdftex-def|xetex-def〉
\def\MT@ifstreq#1#2{\
  \edef\MT@res@a{#1}\
  \edef\MT@res@b{#2}\
  \ifx\MT@res@a\MT@res@b
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
〈pdftex-def|xetex-def〉
\def\MT@ifstreq#1#2{\csname\MT@lua{microtype.if_str_eq([#1],[#2])}\endcsname}
〈luafile〉
local function if_str_eq(s1, s2)
    if s1 == s2 then
        tex_write("@firstoftwo")
    else
        tex_write("@secondoftwo")
    end
end
microtype.if_str_eq = if_str_eq
〈luafile〉
\MT@xadd
Add item to a list.
\MT@xaddb
Add item to the beginning.
\MT@mapclistn \MT@mapclistc \MT@clistfunction \MT@clistbreak
Run \#2 on all elements of the comma list \#1. This and the following is modelled after E\TeX3 commands.
\def\MT@map@clist@list{\MT@map@clist@} 
\let\MT@map@clist@break\@gobble
\def\MT@clist@function{#1}
\MT@map@clist@}

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

\MT@in@clist{\ifMT@inlist@\MT@in@clist\fi}
\MT@in@clist@{\MT@in@clist{#1}\MT@in@clist@}
\MT@in@tlist@{\edef\MT@res@b{#1}\if\MT@res@a\MT@res@b\MT@in@tlist@\fi}
\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}
This is the same as \LaTeX{}'s \verb|\loop|, which we mustn't use, since this could confuse an outer \verb|\loop| in the document.

\verb|\while| Execute \verb|#2| from \verb|#1| up to (excluding) \verb|#2| (much faster than \LaTeX{}'s \verb|\@whilenum|).

\verb|\do| Execute \verb|#1| 256 times, resp. for the whole font for Lua\TeX{}, if loaded by fonts/c/luatex-def.

This is the \verb|lua| function, which is much faster than looping through all glyphs in \TeX{}. Legacy fonts (which this function might be fed with, because fonts/c/luatex-def isn't always getting it right) don't contain a v.index field.
\texttt{thefont = fonts.hashes.identifiers[font.current()]}  
\texttt{end}  
\texttt{if thefont then}  
\texttt{for \texttt{i},\texttt{v} in next,thefont.characters do}  
\texttt{if v.index == nil or v.index > 0 then}  
\texttt{microtype.sprint(["\\@tempcnta=\"]..i..["\relax\MT@do@font\@function"])}  
\texttt{end}  
\texttt{end}  
\texttt{end}  
\texttt{end}  
\texttt{end}  
\texttt{end}  
\texttt{microtype.do_font = do_font}  
\texttt{"/luafile"}  

The \texttt{Xe\TeX} variant.  
\texttt{\xetex-def}  
\texttt{\def\MT@do@font#1\{\@tempcnta=\z@ \MT@loop #1\% \advance\@tempcnta \@one \ifnum\@tempcnta < \XeTeXcountglyphs\MT@font \MT@repeat}}  
\texttt{\}/xetex-def}  

\texttt{\package}  
\texttt{\MT@count}  

Increment macro \texttt{(#1)} by one. Saves using up too many counters. The \texttt{e\TeX} way is slightly faster.  
\texttt{\newcount\MT@count}  
\texttt{\def\MT@increment#1\{\^X \edef#1\{\number\numexpr #1 + 1\relax\}\% \^Q \MT@count=#1\relax \^Q \advance\MT@count \@one \^Q \edef#1\{\number\MT@count\}\% \}}  

\texttt{\MT@scale}  

Multiply and divide a counter. If we are using \texttt{e\TeX}, we will use its \texttt{\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.  
\texttt{\def\MT@scale#1#2#3\{\^Q \multiply #1 #2\relax \ifnum #3 = \z@ \^X \#1=\numexpr #1 * #2\relax \else \^X \#1=\numexpr #1 * #2 / #3\relax \^Q \divide #1 #3\relax \^Q \}}  

\texttt{\MT@abbr@pr}  

Some abbreviations. Thus, we can have short command names but full-length log output.  
\texttt{\MT@abbr@pr\@pr}  
\texttt{\MT@abbr@pr\@ex}  
\texttt{\MT@abbr@pr\@inh}  
\texttt{\MT@abbr@pr\@c}  
\texttt{\MT@abbr@ex\@pr}  
\texttt{\MT@abbr@ex\@ex}  
\texttt{\MT@abbr@ex\@inh}  
\texttt{\MT@abbr@ex\@c}  
\texttt{\MT@abbr@inh\@pr}  
\texttt{\MT@abbr@inh\@ex}  
\texttt{\MT@abbr@inh\@c}  
\texttt{\MT@abbr@inh\@inh}  
\texttt{\MT@abbr@inh\@sp}  
\texttt{\MT@abbr@inh\@sp\@c}  
\texttt{\MT@abbr@kn\@pr}  
\texttt{\MT@abbr@kn\@ex}  
\texttt{\MT@abbr@kn\@inh}  
\texttt{\MT@abbr@kn\@c}  
\texttt{\MT@abbr@kn\@kn}  
\texttt{\MT@abbr@kn\@tr}  
\texttt{\MT@abbr@kn\@tr\@c}
\def\MT@abbr@kn@c{kerning codes}
\def\MT@abbr@kn@inh{kerning inheritance}
\def\MT@abbr@tr{tracking}
\def\MT@abbr@tr@c{tracking amount}

These we also need the other way round.

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

We can work on these lists to save some guards in the \texttt{dtx} file.

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

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

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

14.1.5 Compatibility

For the record, the following \LaTeX kernel commands will be modified by \texttt{microtype}:

* \texttt{\pickup@font}

* \texttt{\do@subst@correction}

* \texttt{\add@accent} (all in section 14.2.9)

* \texttt{\showhyphens} (in section 14.4.6)

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

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

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

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

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

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

\begin{verbatim}
\def\MT@addto@setup{
\g@addto@macro\MT@setup@}
\end{verbatim}

Don't hesitate with \texttt{miniltx}.

\begin{verbatim}
\def\MT@addto@setup{
\g@addto@macro\MT@setup@}
\end{verbatim}

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

\begin{verbatim}
\def\MT@with@package@T#1{% 
\ifMT@inlist@#1\@firstofone\@gobble\fi}
\end{verbatim}

\texttt{LaTeX}'s \texttt{@ifpackage} ignores the class options.

\begin{verbatim}
\def\MT@with@package@T#1{% 
\ifMT@inlist@#1\@firstofone\@gobble\fi}
\end{verbatim}

The \texttt{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 are'n any problems with the \texttt{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 pdfT\TeX\ version 1.21b together with \texttt{ledpatch.sty} as of 2005/06/02 (v0.4), character protrusion will work at last.

\textit{Peter Wilson} was so kind to provide the \texttt{\@dunhbox@line} hook in \texttt{ledmac} to allow for protrusion. \texttt{\leftmarginkern} and \texttt{\rightmarginkern} are new primitives of pdfT\TeX\ 1.21b (aka. 1.30.0). They are also part of recent Xe\TeX. The successor packages \texttt{eledmac} and \texttt{reledmac} are also supported.

\begin{verbatim}
\ifMT@protrusion\MT@ifdefined@c@TF\l@dunhbox@line{ 
\MT@led@unhbox@line
\MT@info{Patching ((r)e)ledmac to enable character protrusion} 
\let\MT@led@unhbox@line\l@dunhbox@line
\renewcommand*{\l@dunhbox@line}[1]{ 
\ifhbox##1% 
\kern\leftmarginkern##1% 
\expandafter\MT@led@unhbox@line\expandafter##1\expandafter 
\kern\rightmarginkern##1% 
\fi}
\fi}
\else
\MT@warning{Character protrusion in paragraphs with \texttt{\line}MessageBreak
\texttt{numbering will only work if you update ledmac,\MessageBreak}
\texttt{or use one of its successors, eledmac or reledmac}}
\fi
\end{verbatim}

\textbf{Hook.}

\begin{verbatim}
\MT@ifdefined@c@TF\\@dunhbox@line%
\end{verbatim}
The `shapepar` package (v2.2) fixes this in a similar manner by itself, so we don't have to bother.

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

The `shapepar` package (v2.2) fixes this in a similar manner by itself, so we don't have to bother.
preamble. However, it is still necessary for \defersetup=false.

\MT@setupfont@hook

\MT@f@false\MT@with@babel@and@T{spanish}\MT@f@true\MT@with@babel@and@T{galician}\MT@f@true\MT@with@babel@and@T{mexican}\MT@f@true\ifMT@if@true\MT@ifdefined@c@T\percentsign{\let\%\percentsign}\fi

Using \disablequotes, we can restore the original meaning of all characters
made active by csquotes. (It would be doable for older versions, too, but we won't
bother.)

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

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

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

\MT@with@package@T{tikz}\MT@tikz@setup

Check again at the end of the preamble.

〈/package〉

\MT@d@t@o@setup{\relax}

〈/package〉

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

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

〈plain〉 \MT@requires@latex2{\MT@with@package@T{fontspec}{\MT@fontspectrue\MT@fontspec@setup}}〈/plain〉

〈plain〉\relax(〈/package〉)

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

\MT@let\MT@setupfont@hook{\empty}

\MT@f@false\MT@with@package@T{galician}\MT@f@true\MT@with@package@T{mexican}\MT@f@true\ifMT@if@true\MT@d@t@o@macro\MT@setupfont@hook{\%\let\%\percentsign}\fi

\MT@with@package@T{csquotes}{\@ifpackagelater{csquotes}{2005/05/11}\disablequotes\relax}
We disable `microtype`’s additions inside `hyperref`’s `pdfstringdef`, which redefines lots of commands. `hyperref` doesn’t work with plain TeX, so in that case we don’t bother.

The `listings` package makes numbers and letters active,

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

Inside a listing, `\space` is redefined.

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

Of course, using both `soul`’s and `microtype`’s letterspacing mechanisms at the same time doesn’t make much sense. But `soul` can do more, e.g., underlining. The optional argument to `\textls` may not be used.

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

\begin{verbatim}
\fontsetup{plain}
\let\SOUL\undefined\else\SOUL\end\let
\fontsetup{textls 1\%}
\let\SOUL\undefined\else\SOUL\end\let
\end{verbatim}

\begin{verbatim}
\MT@with{tikz}{\MT@tikz@setup}
\end{verbatim}

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

\begin{verbatim}
\MT@with{pinyin}{%\let\MT@orig\py@macron
\ifpackagelater{pinyin}{2005/08/11}{%\def\py@macron#1#2{%\MT@ltx@pickupfont\MT@orig\py@macron{#1}{#2}\MT@MT@pickupfont}%\}%;\def\py@macron#1{%\MT@ltx@pickupfont\MT@orig\py@macron{#1}\MT@MT@pickupfont}%\}}%%
\}
\end{verbatim}

\begin{verbatim}
\end{verbatim}

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

\begin{verbatim}
\expandafter\ifx\the\font\nullfont\normalfont\fi
\MT@setupfont
\end{verbatim}

\subsection{Font setup}

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

\begin{verbatim}
\def\MT@setupfont{%\ifpdf\ifxetex\ifluatex\MT@copyfont\relax\fi\fi\relax
\MT@exp\two@c\MT@split\name\string\MT@font/\@nil
\ifx\MT@family\empty\else
\MT@findfile\MT@family\ifx\MT@family\empty\else
\MT@exp\one@n\MT@findfile\MT@family\relax\fi\fi\relax
\fi\fi\fi\fi\fi
\end{verbatim}

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

\begin{verbatim}
\MT@exp\one@n\MT@findfile\MT@font/\@nil\ifx\MT@font\empty\else
\fi\fi\fi\fi\fi
\end{verbatim}
We have to make sure that \cf@encoding expands to the correct value (for later, in \MT@get@slot), which isn’t the case when \selectfont chooses a new encoding (this would be done a second later in \selectfont, anyway – three lines, to be exact). (I think, I do not need this anymore – however, I’m too afraid to remove it. ... Oops, I did it. Let’s see whether anybody complains.)

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

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

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

Interword spacing and kerning (pdf\TeX{} 1.40).

Disable ligatures (pdf\TeX{} 1.30).

Debugging.

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

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

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

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

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

```
\SetExpansion
{ [ stretch = 30,
    shrink = 60,
    step = 5 ]
  [ encoding = *,
    size = 10.001 ] }
\newcommand{\expandpar}[1]{%\fontsize{10.001}{\baselineskip}\selectfont#1\par}%
\expandpar{This paragraph contains an ‘unnecessary’ widow.}
```
Note that the \texttt{\textbackslash expandpar} command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the \texttt{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.

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

\begin{verbatim}
\MT@split@name
\MT@encoding
\MT@family
\MT@series
\MT@shape
\MT@size
\MT@familyalias
\MT@scrubfeature
\MT@scrubfeatures
\MT@checklist
\MT@dotrue
\MT@feature
\MT@featurefalse
\MT@maybefeature
\MT@checklist@font
\MT@checklist@encoding
\MT@checklist@family
\MT@checklist@series
\MT@checklist@shape
\MT@checklist@size
\MT@ifdefined@n@TF{MT@familyalias @alias}{\MT@let@cn\MT@familyalias {MT@family @alias}}{\let\MT@familyalias @empty}
\end{verbatim}

We check all features of the current font against the lists of the currently active font set, and set \texttt{\MT@feature} accordingly.

\begin{verbatim}
begin{Verbatim}
\MT@checklist{font,encoding,family,series,shape,size}{\MT@ifdefined@n@TF{MT@checklist@##1}{\csname MT@checklist@##1\endcsname}{\MT@checklist@{##1}}{#1}}
\end{Verbatim}
\end{verbatim}

\begin{verbatim}
\MT@feature stores the current feature.
\end{verbatim}
\csname MT@set@codes\endcsname
\def\MT@info@notracking0\relax\def\MT@info@notracking@{\MT@vinfo{... No \@nameuse{MT@abbr@#1}}}
\MT@dinfo@list
\MT@checklist@
\MT@checklist@family

\MT@info@notracking
To defer the message to after the font has actually been logged.
\MT@info@notracking0
\def\MT@info@notracking0[\MT@vinfo[... No tracking]]
\MT@dinfo@list
\MT@checklist@
The generic test (\#1) is the axis, (\#2) the feature, \@tempa contains the set name).
\MT@checklist@family
Also test for the alias font, if the original font is not in the list.

\MT@checklist@family
\MT@checklist@size Test whether font size is in list of size ranges.
\MT@checklist@font If the font matches, we skip the rest of the test.

14.2.1 Protrusion
\ifMT@nofamily Info for settings that are not family-specific. (Warnings seem to be too irritating.)

The switch is set in \MT@next@listname.

\MT@protrusion Set up for protrusion?

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

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

1178 \MT@if@list@exists{%}
1179 \ifMT@nofamily
1180 \MT@ifdef@onMTF\MT@encoding-\MT@family-settings\relax{%
1181 \MT@info@nl{Loading generic protrusion settings for font family \MT@family' (encoding: \MT@encoding). MessageBreak}
1182 For optimal results, create family-specific settings. MessageBreak
1184 See the microtype manual for details}%
1185 \MT@glet@nc{\MT@encoding-\MT@family-settings}\@empty
1186 }%
1187 \fi
1188 \MT@get@font@dimen@six{%
1189 \MT@get@opt
1190 \MT@get@inh@list

Set an input encoding?

1192 \MT@set@inputenc{c}%

Load additional lists?

1193 \MT@load@list\MT@pr@c@name
1194 \MT@set@listname

Load the main list.

1195 \MT@let@cn@tempc{MT@pr@c\MT@pr@c@name}%
1196 \expandafter\MT@set@codes@tempc,\relax,}%
1197 }\MT@reset@pr@codes

1198 }

\MT@get@font@dimen@six
\MT@dimen@six

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

1199 \def\MT@get@font@dimen@six{%
1200 \ifnum\fontdimen6\MT@font=\z@
1201 \MT@warning@nl{Font `\MT@@font' does not specify its MessageBreak}
1202 \@backslashchar fontdimen 6 (width of an `em')! Therefore, MessageBreak
1203 \@nameuse{MT@abbr@MT@feat} will not work with this font}%
1205 \expandafter\@gobble
1206 \else
1207 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
1208 \expandafter\@firstofone
1209 \fi
1210 }

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

1211 \def\MT@set@all@pr#1#2{%
1212 \let\MT@temp\@empty
1213 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp\lpcode\MT@font\@tempcnta=#1}%
1215 \MT@tempcnta=#2}%
1216 \g@addto@macro\MT@font\tempcnta=#2}%
1217 \MT@do@font\MT@temp}

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

1218 \def\MT@reset@pr@codes0{\MT@set@all@pr\z@\z@}
1219 \let\MT@reset@pr@codes\relax
If the font is letterespaced, we have to add half the letterspacing amount to the margin kerns. This will be activated in \MT@set@tr@codes.

\MT@get@char@unit may mean different things.

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

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 ‘fi’ ligature). Unlike protcode.tex and pdfcprot, we do not calculate with \lp\textup{pcode} resp. \rp\textup{pcode}, since this would disallow protrusion factors larger than the character width (since \texttt{\[lr\]pcode}'s limit is 1000). Now, the maximum protrusion is 1em of the font.

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

\begin{verbatim}
〈pdftex-def〉\MT@requires@pdftex3 {\def\MT@scale@to@em{\@tempcntb=\MT@count \relax} \def\MT@scale@factor{\ifnum\@tempcntb=\z@ \else \MT@scale\@tempcntb \@tempb \MT@dimen@six \fi}}〈/pdftex-def〉
\end{verbatim}

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

\begin{verbatim}
\MT@scale\@tempcntb \@tempb \MT@dimen@six \ifnum\@tempcntb=\z@ \else \MT@scale@factor \fi
\end{verbatim}

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

\begin{verbatim}
〈pdftex-def〉\def\MT@get@charwd{\MT@count=\fontcharwd\MT@font\MT@char \relax}〈/pdftex-def〉〈xetex-def〉\MT@get@charwd\MT@count=\fontcharwd\MT@font\MT@char \relax
\end{verbatim}

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

\begin{verbatim}
〈xetex-def〉\ifnum\MT@char<\z@ \setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char} \MT@count=\wd\z@ \else \MT@count=\fontcharwd\MT@font\MT@char \relax \fi
〈/xetex-def〉
〈/pdftex-def〉
\end{verbatim}

\begin{verbatim}
\ifnum\MT@count=\z@ \MT@info@missing@char \fi
\end{verbatim}

For letterspaced fonts, we have to subtract the letterspacing amount from the characters' widths. The protrusion amounts will be adjusted in \texttt{\MT@set@pr@codes}.

\texttt{\MT@char} is already loaded so that 1em = \texttt{\fontdimen6}.

\begin{verbatim}
〈pdftex-def〉\MT@requires@pdftex6{\g@addto@macro\MT@get@charwd{\MT@ifdefined@c@T\MT@letterspace{\advance\MT@count -\dimexpr\MT@letterspace sp *\dimexpr 1em/1000\relax}}}〈/pdftex-def〉
\end{verbatim}

No adjustment with versions 0.14f and 0.14g.
We need this in \MT@warn@code@too@large (neutralised).

\MT@font@dimen

For the space unit.

\MT@info@missing@char

Info about missing characters, or characters with zero width.

\MT@scale@factor

Furthermore, we might have to multiply with a factor.

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

\MT@get@opt

\MT@pr@factor@ Apply a factor?

\MT@kn@factor@ 1356 \def\MT@get@opt{\MT@set@listname

\MT@sp@factor@ 1357 \MT@pr@factor@ \MT@kn@factor@

\MT@kn@unit@ 1358 \MT@ifdefined@n@TF{MT@MT@feat@c@csname MT@MT@feat@c@endcsname \MT@feat@factor}{% 1359 \MT@let@nn{MT@MT@feat@factor}{MT@MT@feat@factor}

\MT@kn@unit@ 1360 }{MT@MT@feat@factor@}(MT@MT@feat@factor)% 1361 \MT@vinfo{... : Multiplying \nameuse{MT@abbr@MT@feat} codes by 1362 \number\csname MT@MT@feat@factor@endcsname/1000}% 1363 }{% 1364 \MT@let@nn{MT@MT@feat@factor}{MT@MT@feat@factor}

\MT@pr@unit@ 1365 }% 1366 \MT@get@space@unit 1367 \MT@get@char@unit

\MT@pr@unit@ The unit can only be evaluated here, since it might be font-specific. If it's \@empty, 1368 it's relative to character widths, if it's −1, relative to space dimensions.

\MT@kn@unit@ 1369 \MT@ifdefined@n@TF{MT@MT@feat@c@csname MT@MT@feat@c@endcsname \MT@feat@unit}{% 1370 \MT@let@nn{MT@MT@feat@unit}{MT@MT@feat@unit}

\MT@kn@unit@ 1371 }{MT@MT@feat@unit}(MT@MT@feat@unit)% 1372 }% 1373 \let\MT@get@char@unit\relax 1374 \let\MT@get@space@unit\@gobble 1375 \MT@exp@cs\ifx{MT@MT@feat@unit}@\@empty% 1376 \let\MT@get@char@unit\MT@get@charwd

\MT@kn@unit@ 1377 \else 1378 \MT@exp@cs\ifx{MT@MT@feat@unit}@\m@ne% 1379 \let\MT@get@space@unit\MT@get@font@dimen

\MT@kn@unit@ 1380 \else 1381 \MT@exp@cs\MT@get@unit{MT@MT@feat@unit}% 1382 \fi 1383 \fi 1384 \fi 1385 \fi 1386 \fi

\MT@get@unit

\MT@get@unit@

\MT@get@unit@ The codes are either relative to character widths, or to a fixed width. For spacing 1381 and kerning lists, they may also be relative to the width of the interword glue. Only 1382 the setting from the top list will be taken into account.

\MT@get@unit@ 1383 \let\MT@get@char@unit\relax 1384 \let\MT@get@space@unit\@gobble 1385 \MT@exp@cs\ifx{MT@MT@feat@unit}@\@empty% 1386 \let\MT@get@char@unit\MT@get@charwd

\MT@get@unit@ 1387 \else 1388 \MT@exp@cs\ifx{MT@MT@feat@unit}@\m@ne% 1389 \let\MT@get@space@unit\MT@get@font@dimen

\MT@get@unit@ 1390 \else 1391 \MT@exp@cs\MT@get@unit{MT@MT@feat@unit}% 1392 \fi 1393 \fi 1394 \fi 1395 \fi

\MT@get@unit 1396

\MT@get@unit@ If unit contains an em or ex, we use the corresponding \fontdimen to obtain the 1397 real size. Simply converting the em into points might give a wrong result, since 1398 the font probably isn't set up yet, so that these dimensions haven't been updated, 1399 either.

\MT@get@unit@ 1390 \let\MT@get@char@unit\relax 1391 \let\MT@get@space@unit\@gobble 1392 \MT@exp@cs\ifx{MT@MT@feat@unit}@\@empty% 1393 \let\MT@get@char@unit\MT@get@charwd

\MT@get@unit@ 1394 \else 1395 \MT@exp@cs\ifx{MT@MT@feat@unit}@\m@ne% 1396 \let\MT@get@space@unit\MT@get@font@dimen

\MT@get@unit@ 1397 \else 1398 \MT@exp@cs\MT@get@unit{MT@MT@feat@unit}% 1399 \fi 1400 \fi 1401 \fi 1402 \fi

\MT@get@unit@ 1403 \def\MT@get@unit@#1{% 1404 \expandafter\MT@get@unit@#1 e!@nil

\MT@get@unit@ 1405 \ifx\empty\MT@get@unit@\MT@get@unit@\relax\@nil

\MT@get@unit@ 1406 \defaultunits\@tempdima pt\relax\@nil
The configurations may be under the regime of an input encoding.

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

More recent versions of inputenc remember the current encoding, so that we can test whether we really have to load the encoding file.

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

Set the inheriting characters.
\MT@preset@pr
\MT@preset@pr@ Preset characters. Presetting them relative to their widths is not allowed.

\MT@preset@aux
\MT@preset@aux@factor
\MT@preset@aux@space AUXILIARY MACRO FOR PRESETTING. STORE VALUE \#1 IN MACRO \#2.

\MT@warn@preset@towidth
\MT@warn@preset@towidth\@pr
\MT@warn@preset@towidth\@factor
\MT@warn@preset@towidth\@space CAN'T PRESET CHARACTERS RELATIVE TO THEIR WIDTHS. MESSAGE BREAK.

\MT@expansion
\MT@expansion\@factor
\MT@expansion\@space SET UP FOR EXPANSION?

\MT@set@ex@codes@s
\MT@set@ex@codes@s\@factor
\MT@set@ex@codes@s\@space SETTING UP FONT EXPANSION IS A BIT DIFFERENT BECAUSE OF THE SELECTED OPTION. THERE ARE TWO VERSIONS OF THIS MACRO.

14.2.2 Expansion
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
\else
If \texttt{MT@nonselected} is \texttt{false}, we need this boolean in \texttt{MT@if@list@exists} so that no warning for missing lists will be issued.
\fi

\MT@set@ex@codes
\MT@set@all@ex
\MT@reset@ef@codes
\MT@reset@ef@codes\mt@ex@codes\n
\MT@set@all@ex\number#1
\MT@reset@ef@codes
\MT@reset@ef@codes

\MT@reset@ef@codes\mt@ex@codes\n
Expand the font.

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

However, this is only necessary for pdfTeX versions prior to 1.20, or Lua\TeX{} < 0.90 (actually, I think, 0.87).
\MT@ex@split@val

There's only one number per character.
\def\MT@ex@split@val#1\relax{%  
\@tempcntb=#1\relax
  \ifnum\MT@ex@factor@=\@m\else
    \MT@scale\@tempcntb \MT@ex@factor@ \@m
  \fi
  \ifnum\@tempcntb > \MT@ex@max
    \MT@warn@ex@too@large\MT@ex@max
  \else
    \ifnum\@tempcntb < \MT@ex@min
      \MT@warn@ex@too@large\MT@ex@min
    \fi
  \fi
  \efcode\MT@font\MT@char=\@tempcntb
  \〈debug\MT@dinfo@nl{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%}

Heirs, heirs, I love thy heirs.
\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

Apply different values to this font?
\MT@ex@factor@
\MT@get@ex@opt{%  
\def\MT@get@ex@opt\relax{%  
\MT@get@list@name\MT@ifdefined@n@TF{MT@ex@c@name factor}{%  
\MT@let@cn\MT@ex@factor{MT@ex@c@name factor}%
\MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor/1000}%
\else
  \MT@let@nn{MT@ex@factor}{MT@ex@factor}%
\fi
%  }
%  }
%  }
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
%}
\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}% 
} 
\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 
\MT@if@list@exists{% 
\MT@get@font@dimen@six{% 
\MT@get@opt 
\MT@reset@sp@codes 
\MT@get@inh@list 
\MT@set@inputenc{c}% 
\MT@load@list\MT@sp@c@name 
\MT@set@listname 
\MT@let@cn\@tempc{MT@sp@c@\MT@sp@c@name}% 
\expandafter\MT@set@codes\@tempc,\relax,}% 
\MT@reset@sp@codes 
}\MT@reset@sp@codes 
\MT@sp@split@val 
If \texttt{unit=space}, \MT@get@space@unit will be defined to fetch the corresponding
fontdimen (2 for the first, 3 for the second and 4 for the third argument).
\def\MT@sp@split@val#1,#2,#3\relax{% 
\def\@tempb{#1}% 
\MT@ifempty\@tempb\relax{% 
\MT@get@space@unit2% 
\MT@scale@to@em 
\knbscode\MT@font\MT@char=\@tempcntb 
〈debug〉\MT@dinfo@nl{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
} 
\def\@tempb{#2}% 
\MT@ifempty\@tempb\relax{% 
\MT@get@space@unit3% 
\MT@scale@to@em 
\stbscode\MT@font\MT@char=\@tempcntb 
〈debug〉\MT@dinfo@nl{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
} 
\def\@tempb{#3}% 
\MT@ifempty\@tempb\relax{% 
\MT@get@space@unit4% 
\MT@scale@to@em 
\shbscode\MT@font\MT@char=\@tempcntb 
〈debug〉\MT@dinfo@nl{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
} 
\MT@ifdefined@c@T\MT@sp@inh@name{%
\MT@ifdefined@n@T{MT@inh@\MT@sp@inh@name @\MT@char @}{%
14.2.4 Additional kerning

It's getting boring, I know.

Again, only check for additional kerning for new versions of pdfTeX.
Again, the unit may be measured in the space dimension; this time only \fontdimen 2.

\def\MT@kn@split@val#1,#2\relax{% 
\def\@tempb{#1} 
\MT@ifempty\@tempb\relax{% 
\MT@get@space@unit2 
\MT@scale@to@em 
\knbccode\MT@font\MT@char=\@tempcntb 
\MT@do@font\@tempcntb 
\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char} 
\MT@ifdefined@c@T\MT@kn@inh@name{% 
\MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{% 
\MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs 
\MT@dinfo@nl{2}{-- heir of \MT@char: #1} 
\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char} 
\MT@ifempty{#1}\relax{}{\MT@preset@aux{#1}\@tempa} 
\MT@ifempty{#2}\relax{}{\MT@preset@aux{#2}\@tempb} 
\MT@set@all@kn\@tempa\@tempb 
\MT@ifdefined@c@T\MT@kn@inh@name{% 
\MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{% 
\MT@warn@preset@towidth{kn} 
\MT@do@font\@tempcntb 
\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char} 
\MT@dinfo@nl{2}{-- heir of \MT@char: #1} 
\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char} 
\MT@ifempty{#1}\relax{}{\MT@preset@aux{#1}\@tempa} 
\MT@ifempty{#2}\relax{}{\MT@preset@aux{#2}\@tempb} 
\MT@set@all@kn\@tempa\@tempb 
\MT@ifdefined@c@T\MT@kn@inh@name{% 
\MT@ifdefined@n@T{MT@inh@\MT@kn@inh@name @\MT@char @}{% 
\MT@exp@cs\MT@map@tlist@c{MT@inh@\MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs 
\MT@dinfo@nl{2}{-- heir of \MT@char: #1} 
\MT@dinfo@nl{4}{;;; knbc (#1): \number\knbccode\MT@font\MT@char} 
\MT@ifempty{#1}\relax{}{\MT@preset@aux{#1}\@tempa} 
\MT@ifempty{#2}\relax{}{\MT@preset@aux{#2}\@tempb} 
\MT@set@all@kn\@tempa\@tempb
14.2.5 Tracking

This only works with pdfTEX 1.40 or LuaTEX 0.62.

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

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

Zero tracking requires special treatment.

Letterspacing only works in PDF mode.

The letterspaced font instances are saved in macros \(\langle font\ name\rangle / \langle letterspacing\ amount\rangle\) ls.

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

In case of nested letterspacing with different amounts, we have to extract the base font again.
luatools provides the faux font feature kernfactor, which we will use when dealing with non-legacy fonts, as it is less problematic and faster than the pdfTEX primitive \letterspace.

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

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

\begin{verbatim}
\usepackage{fontspec}
\setkernfactor{}\setletterspace{1}\setletterspace{2}\setletterspace{3}
\end{verbatim}

\texttt{luaotfload} provides the faux font feature \texttt{kernfactor}, which we will use when dealing with non-legacy fonts, as it is less problematic and faster than the pdfTEX primitive \texttt{\letterspace}.
Adjust surrounding spacing and kerning.

We get the current outer spacing and adjust it, then, after the end of the current outer group, set the current outer spacing, again, and adjust.

\MT@set@curr@os

\ifx\MT@ls@adjust\@empty
\MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax
\else
\MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
\csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
\ifdim\MT@outer@kern=\z@else \MT@ls@outer@k \fi
\expandafter\expandafter\expandafter\@secondoftwo
\csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
\fi

\MT@set@curr@ok

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

\MT@afteraftergroup

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

(Following an idea of Will Robertson.)
Add the kernfactor feature to a font loaded by fonts Lipsum (we might have to add the colon ourselves).

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

Various settings (only for the microtype version).

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

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

```
\MT@set@lsfont
```

Adjust interword spacing.

```
\MT@tr@ispace
```

Adjust outer kerning.

```
\MT@tr@okern
```

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

```
\MT@tr@ligatures
```

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

```
\MT@set@lsfont
```
\lsstyle

Disable the tests whether the font should be letterspaced, then trigger the setup. Only \textls can be used in math mode (\lsstyle may be used inside another text switch, of course). Still, we have to ensure that math fonts are set up again. Setting \glob@curr@size to \@empty (our previous solution) could throw us into an infinite loop (e.g., with the psnfss packages, via \every@math@size), so we issue \glob@settings instead.

\DeclareRobustCommand\lsstyle{%
\not@math@alphabet\lsstyle\textls
〈pdftex-def\luatex-def〉\MT@maybe@gobble@with@tikz\aftergroup\glob@settings\%
〈pdftex-def\luatex-def〉\def\MT@feat{tr}\
\let\MT@tracking\MT@set@tr@codes
\selectfont
}

Now the definitions for the \textsc{letterspace} package with plain \TeX.

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

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

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

PDF\TeX cannot letterspace fonts that already are letterspaced. Therefore, we have to save the base font in \font@name@base.

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

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

If tracking is switched off in the middle of the document, or if \textls is called with a zero letterspacing amount, we have to retrieve the base font and select it.
\MT@noligatures \pdfTeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

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

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

\MT@tr@set@space0 If settings for outer spacing (#2) don’t exist, they will be inherited from the inner spacing settings (#1).
If the value is followed by an asterisk, the fontdimen will be scaled by the respective amount, otherwise the value denotes the desired dimension in the respective unit.

\MT@tr@set@space@@ If the value is followed by an asterisk, the fontdimen will be scaled by the respective amount, otherwise the value denotes the desired dimension in the respective unit.

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

\MT@tr@outer@l Disable left outer kerning.

The ragged2e package sets \spaceskip without glue.
microtype also adjusts spacing. The following is borrowed from soul. I've added the cases for italic correction, since tracking may also be triggered by text commands (e.g., \textsc).

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

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

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

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

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

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

If the next token is \check@icr (from an inner text command), we insert ourselves just before it. This will then call \maybe@ic again the next round (which however will always insert an italic correction, since it doesn't read beyond our group).
\MT@if@outer@next@\xobeysp\relax{%  
\MT@if@outer@next\xspace{%  
\def\MT@temp*\xspace{\MT@xspace}  
\def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k\fi}\MT@let@nc{MT@tr@outer@next}\relax  
}  
\fi}\fi\MT@temp*}%
\MT@tr@outer@icr
\MT@tr@outer@icr@
\MT@xspace\MT@xspace@
\MT@ls@adjust@
\textls

xspace requires special treatment.

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

\def\MT@tr@outer@icr\afterassignment\MT@tr@outer@icr0\MT@tr@outer@r  
\def\MT@tr@outer@icr0{%  
\let\@let@token= \MT@tr@outer@next  
\maybe@ic@  
}\MT@xspace\MT@xspace0
\MT@ls@adjust@
\textls

For older pdfTeX versions and LuaTeX, throw an error.

\DeclareRobustCommand\lsstyle{%  
\MT@error{Letterspacing only works with \MT@engine tex version  
\pdftex-def\lualatex-def\MessageBreak or newer}\MT@glet\lsstyle\relax  
}

And for XeLaTeX, too.

\pdftex-def\lualatex-def
\DeclareRobustCommand\lsstyle{%  
\MT@error{Letterspacing currently doesn't work with xetex}\MT@glet\lsstyle\relax  
}

\textls\MT@ls@adjust0
This command may be used like the other text commands. The starred version removes kerning on the sides. The optional argument changes the letterspacing factor.

\ifstar{\MT@ls@adjust0\MT@ls@adjust0\MT@textls}
This is now almost \LaTeX's \DeclareTextFontCommand, with the difference that we adjust the outer spacing and kerning also for \lsstyle, while \LaTeX's text switches don't bother about italic correction.

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

\MT@ls@set@ls{#1}{%  
\KV@@sp@def\MT@letterspace@{#1}%  
\edef\MT@letterspace@{\number\MT@letterspace@}%  
\MT@ls@too@large\MT@letterspace@}

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

\MT@tr@set@okern{#1,#2,}{%  
\let\MT@temp\@empty
\MT@ifempty{#1}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#1}}%
\MT@ifempty{#2}{\MT@tr@set@okern@{*}}{\MT@tr@set@okern@{#2}}%
\MT@glet@nc{MT@outer@kern\expandafter\string\font@name}\MT@temp
\MT@dinfo@nl2{... outer kerning: (#1,#2) = \@nameuse{MT@outer@kern\expandafter\string\font@name}}%
}

\MT@tr@set@okern@#1{%  
\MT@test@ast#1*\@nil{%
\MT@ifdefined@c\TF\MT@tr@unit@{\edef\@tempb{#1}\MT@scale@to@em}{\@tempcntb=#1\relax}%
\MT@ifempty{#1}{{\MT@tr@set@okern@{\@empty}}}{\MT@tr@set@okern@{#1}}%
\MT@ifempty{#2}{{\MT@tr@set@okern@{\@empty}}}{\MT@tr@set@okern@{#2}}%
\MT@glet@nc{MT@outer@kern\expandafter\string\font@name}\MT@temp
\MT@dinfo@nl2{... outer kerning: (#1,#2) = \@nameuse{MT@outer@kern\expandafter\string\font@name}}%
}
\MT@ls@outer@k Adjust outer kerning. We additionally add a marker (\kern3sp\kern-3sp) for cases of nested letterspacing without anything actually printed.

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

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

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

No ‘inputenc’ key.
With LuaTeX, we additionally register the ligatures that should be inhibited in a table (used by the luatexload function keepligature).

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

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

```lua
local function keepligature(c)
    local f = getfont(c)
    local f, ch
    if type(c) == "userdata" then -- in older luatexload versions, c was a node
        f = c.font
        ch = c.components.char
    else -- since 2.6, c is a (direct node) number
        f = getfont(c)
        ch = getfield(getfield(c, "components"), "char")
    end
    if f == tex.fontidentifier(f) then -- should always be true
        local ligas = microtype.ligs[match(tex.fontidentifier(f), "([^ ]+)"
        if ligas then
            for _, liga in ipairs(ligas) do
                if liga == "_all_" or tonumber(liga) == ch then
                    return false
                end
            end
            return true
        end
    end
return true
```
if luaotfload and luaotfload.letterspace then
    if luaotfload.letterspace.keepligature then
        microtype.warning("overwriting function `keepligature'")
    end
else
    luaotfload.letterspace.keepligature = keepligature
end

14.2.7 Loading the configuration

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

\MT@find@file
Micro-typographic settings may be written into a file \texttt{mt-\langle font family \rangle.cfg}.
We must also record whether we've already loaded the file.

Check for existence of the file only once.

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@cfg@catcodes We have to make sure that all characters have the correct category code. Especially, new lines and spaces should be ignored, since files might be loaded in the middle of the document. This is basically \fss@catcodes (from the \LaTeX{} kernel). I’ve added: \& (in \tabular{}s), \!, ?, ; : (french), ,, $, _, ~, and = (Turkish babel).

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

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

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

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

\MT@get@basefamily The family name might have a suffix e.g., for expert set (x), old style numbers (j) swash capitals (w) etc. We mustn’t simply remove the last letter, as this would make for instance cms out of cmss and cmsy (OK, cmex will still become cme …). We only work on the font name if it is longer than three characters.
Table 4: Order for matching font attributes

<table>
<thead>
<tr>
<th></th>
<th>1.</th>
<th>2.</th>
<th>3.</th>
<th>4.</th>
<th>5.</th>
<th>6.</th>
<th>7.</th>
<th>8.</th>
<th>9.</th>
<th>10.</th>
<th>11.</th>
<th>12.</th>
<th>13.</th>
<th>14.</th>
<th>15.</th>
<th>16.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Series</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Shape</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>

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

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

\MT@try@order Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don’t need table 4 in the documentation part any longer and can cast it off here.

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

Also try with an alias family.

\MT@try@order
\MT@next@listname

\MT@get@listname@
\MT@familyalias
\ifnumf=\ifnumg\MT@series\fi
\ifnumh=\MT@shape\fi
\ifnumi=\MT@context\%\fi
\MT@dinfo@nl{1}{(alias) \@tempa}\%\fi
\MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%\fi
\MT@next@listname@#4\%\fi
\fi
\MT@next@listname@If size is to be evaluated, do that, otherwise use the current list.
\MT@context\def\MT@if@list@exists{\MT@let@cn\MT@context{MT@\MT@feat @context}\MT@ifstreq@\MT@context{\MT@let@nc{MT@\MT@feat @c@name}\@empty}\relax\MT@get@listname@\MT@context}{\MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}\MT@ifdefined@c@TF\MT@listname{\MT@vinfo{... Applying non-selected expansion (list `\MT@listname')}\if\MT@nonselected\MT@vinfo{... Applying non-selected expansion (no list)}\else\MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\MT@listname'}\fi\MT@firstoftwo\MT@secondoftwo\MT@get@inh@list\MT@context}{\MT@context}{Since the name cannot be \@empty, this is a sound proof that no matching list exists.\MT@context}{\MT@let@nc{MT@\MT@feat @c@name}\@empty\MT@context}{Don’t warn if selected=false.\MT@context}{\MT@vinfo{... Applying non-selected expansion (no list)}\else\MT@vinfo{... Applying non-selected expansion (list `\MT@listname')}\fi\MT@firstoftwo\MT@secondoftwo\MT@get@inh@list\MT@context}{The inheritance lists are global (no context).\MT@context}{\MT@let@n@T{MT@\MT@context}{\MT@context}{\MT@context}{}}
14.2.8 Translating characters into slots

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

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.

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char

\MT@char@

\MT@get@slot

\MT@char
• Now, we’ll catch the rest, which hopefully is an accented character (e.g. ´a).

• It could also be a \chardef command (e.g., the percent character). This seems the least likely case, so it’s last.

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

In LuaTEX, it may also be a glyph name, prefixed with ‘/’.

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.

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

\begin{verbatim}
\ifnum\MT@char=47\relax
\if\MT@norest \edef\MT@char{U47}\% \else
\@tempcnta=\XeTeXglyphindex\expandafter\@gobble\@tempa\relax
\ifnum\@tempcnta=\z@ \MT@warn@unknown
\let\MT@char=\@empty
\else
\edef\MT@char{(\@tempa}\space}
\edef\MT@char@{-\the\@tempcnta}\%
\endgroup
\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}\%
\fi
\fi
\ifnum\MT@char>\m@ne
\if\MT@norest
\edef\MT@char{U\MT@char}\%
\else
\MT@info@missing@char
\let\MT@char=\@empty
\fi
\else
\MT@warn@rest
\let\MT@char=\@empty
\fi
\else
\MT@warn@unknown
\let\MT@char=\@empty
\fi
\fi
\fi
\endverbatim

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

\begin{verbatim}
\edef\MT@char\@empty
\ifnum\MT@char<\m@ne
\MT@warn@unknown
\let\MT@char=\@empty
\else
\MT@info@missing@char
\let\MT@char=\@empty
\fi
\fi
\endverbatim

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

\begin{verbatim}
if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
local slot_of_name = luaotfload.aux.slot_of_name
microtype.name_to_slot = function(name, unsafe)
return slot_of_name(font.current(), name, unsafe)
end
else
-- we dig into internal structure (should be avoided)
local function name_to_slot(name, unsafe)
if fonts then
local unicodes
if fonts.ids then
local tfmdata = fonts.ids[font.current()]
elseif not tfmdata then return end
unicodes = tfmdata.shared.otfdata.luatex.unicodes
else
local tfmdata = fonts.hashes.identifiers[font.current()]
end
\end{verbatim}
IMPLEMENTATION: Font setup

2633  unicodes = tfmdata.resources.unicodes
2634  end
2635  local unicode = unicodes[name]
2636  if unicode then --- does the 'or' branch actually exist?
2637    return type(unicode) == "number" and unicode or unicode[1]
2638  end
2639  end
2640  end
2641  microtype.name_to_slot = name_to_slot
2642  end
2643
2644  \MT@is@letter  Input is a letter, a character or a number.
2645  \MT@max@char  Warning if resulting character or slot number is too large.
2646  \MT@max@slot  Test whether all of the string has been used up.
2647
2648  \def\MT@is@number#1#2#3\relax{\if\relax#3\relax\else
2649    \if\relax#2\relax\else
2650      \MT@noresttrue
2651      \if#1"\relax
2652        \edef\MT@char@{\number#1#2#3}
2653        \MT@dinfo@nl{3}{> ... a hexadecimal number: \MT@char@}
2654      \else
2655        \MT@norestfalse
2656      \fi
2657    \fi
2658  \fi
2659  \fi
2660  \fi
2661
2662  \newif\ifMT@norest
2663  \ifcat a\noexpand#1\relax%  \edef\MT@char@{\number`#1}%
2664  \ifx\relax#2\relax%
2665  \ifcat !\noexpand#1\relax%
2666  \edef\MT@char@{\number`#1}%
2667  \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
2668  \MT@norestfalse
2669  \fi
2670  \expandafter\MT@is@number#1#2#3\relax\relax
2671  \fi
2672  \fi
2673  \fi
2674
2675  
2676
2677  Numbers may be specified as a three-digit decimal number (029), as a hexadecimal
2678  number (prefixed with "{:1D} or as a octal number (prefixed with '{:35). They
2679  must consist of at least three characters (including the prefix), that is, "F is not
2680  permitted.
2681  \def\MT@is@number#1#2#3\relax{}%  \edef\MT@char@{\number#1#2#3}%
2682  \if\relax#3\relax\else
2683  \if\relax#2\relax\else
2684  \MT@noresttrue
2685  \if#1"\relax
2686  \edef\MT@char@{\number#1#2#3}%
2687  \fi
2688  \fi
2689  \fi
2690  \fi
2691  \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., Å into "A, that is
to whatever it is defined in the inputenc encoding file.

Unfortunately, the (older) inputenc definitions prefer the protected/generic
variants (e.g., \copyright instead of \textcopyright), which our parser won't
be able to understand. (I'm fed up now, so you have to complain if you really,
really want to be able to write '©' instead of \textcopyright, thus rendering your
configuration files unportable.)

Unicode characters (inputenc/utf8,utf8x) are also supported.

\MT@is@symbol

The symbol commands might expand to funny stuff, depending on context. Instead
of simply expanding \(\text{\textbackslash command}\), we construct the command
\(\text{\textbackslash encoding\textbackslash command}\) and see whether its meaning is \(\text{\textbackslash char"\text{"\text{hex number}}\), which is the case for everything
that has been defined with \texttt{\textbackslash DefineTextSymbol} in the encoding definition files.
... or, if it hasn't been defined by \DeclareTextSymbol, a letter (e.g., \texttt{i}, when using frenchpro).

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

With fontspec's TU encoding, glyph numbers may be up to four digits.

For xunicode, which doesn't \countdef, but rather \defs the chars.

Here, we are dealing with accented characters, specified as two tokens.
Again, we construct a control sequence, this time of the form: `\(\langle encoding\rangle\langle\text{accent}\rangle-\langle character\rangle\)`, e.g., `\(\langle T1\rangle-\langle a\rangle\)`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\stringifying` it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

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

\begin{verbatim}
\ifx\UnicodeEncodingName\@undefined\else
  \expandafter\expandafter\expandafter\MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
\fi
\expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
\end{verbatim}

Again, xunicode.

\begin{verbatim}
\ifnum\MT@char@ < \z@
  \ifMT@xunicode
    \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}\
    \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter\meaning\expandafter\x\mt@mathcharstring\relax\relax\relax\relax\relax
  \fi
\fi
\fi
\end{verbatim}

[What about math? Well, for a moment the following looked like a solution, with `\mt@is@mathchar` defined accordingly, analogous to `\MT@is@char` above, to pick up the last two tokens (the `\meaning` of a `\mathchardef`ed command expands to its hexadecimal notation):

\begin{verbatim}
\def\MT@is@mathchar#1{
  \if\relax\noexpand#1
    \let\x#1
  \else
    \mathchardef\x=\mathcode`#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter\meaning\expandafter\x\mt@mathcharstring\relax\relax
}
\end{verbatim}

However, the problem is that `\mathcodes` and `\mathchardefs` have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e.g., the minus in `cmsy` when the `euler` package is loaded). So, no way to go, unfortunately.]

Some warning messages, for performance reasons separated here.

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

For ‘other’ characters > 127, we issue a warning (inputenc probably hasn’t been loaded), since correspondence with the slot numbers would be purely coincidental.

\MT@warn@number@too@large 

Number too large.

\MT@warn@rest 

Not all of the string has been parsed.

\MT@warn@unknown 

No idea what went wrong.

\MT@warn@maybe@inputenc 

In case an input encoding had been requested.

14.2.9 Hook into \LaTeX{}’s font selection

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

In contrast to the pdfcprot package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up.

For my reference:

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

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

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

\MT@font@list
\MT@font
\MT@orig@pickupfont
\MT@requires@latex2{%

We use a comma separated list.

\MT@font@list
\MT@font
\let\MT@font@list\@empty
\let\MT@font\@empty
All this is done at the beginning of the document. It doesn't work for plain, of course, which doesn't have \pickup@font.

\MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
\MT@with@package@T{xeCJK} {\MT@warn@unknown@once{xeCJK}}%

\MT@orig@pickupfont

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

\MT@with@package@T{luatexja}{\MT@warn@unknown@once{luatexja}}%
\MT@with@package@T{xeCJK} {\MT@warn@unknown@once{xeCJK}}%

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

\MT@requires@latex2{%

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

\MT@with@package@T{xeCJK}{\MT@requires@latex2{%

\MT@orig@pickupfont
}\MT@with@package@T{xeCJK}{\MT@warn@unknown@once{xeCJK}}%
CJKutf8 redefines \pickup@font once more (recent versions, in PDF mode, as determined by ifpdf, which CJKutf8 loads).

\ifpackage{CJKutf8}
\ifpackage{CJKutf8}{2008/05/22} 4.8.0
\ifpdf\expandafter\ifexpandafter\else\expandafter\ifexpandafter[0firstoftwo]\fi\fi
\@ifpackageloaded{CJKutf8}\%\fi
\@ifpackagelater{CJKutf8}{2008/05/22}\%4.8.0\fi
\ifpdf\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi\%
\@firstoftwo\%
\@firstoftwo\%
\g@addto@macro\MT@orig@pickupfont{%\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax\define@newfont\else\xdef\font@name{\csname\curr@fontshape/\f@size/\CJK@plane\endcsname}}\%
\g@addto@macro\MT@orig@pickupfont{%\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax\define@newfont\def\CJK@temp{v}\ifx\CJK@temp\CJK@plane\expandafter\ifx\csname CJK@cmap@\f@family\CJK@plane\endcsname\relax\else\csname CJK@cmap@\f@family\CJK@plane\endcsname\fi\else\CJK@addcmap\CJK@plane \fi\else\expandafter\if\font@name\relax\define@newfont\fi\fi\}
\@gobble
\@gobble
\}{{0firstofnone}\%}

This is the normal \LaTeX{} definition.
\ifx\pickup@font\MT@orig@pickupfont\else
\MT@warning@nl{Command \string\pickup@font space is not defined as expected.\MessageBreak Patching it anyway. Some things may break\MessageBreak \langle\package\rangle.\MessageBreak Double-check whether micro-typography is indeed\MessageBreak applied to the document.\MessageBreak (Hint: Turn on `verbose' mode)\MessageBreak \langle\package\rangle.\fi

\pickup@font

Then we append our stuff. Everything is done inside a group.
\g@addto@macro{\pickup@font}{\begingroup}

If the trace package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.
\MT@with@package{trace}{\@g@addto@macro{\pickup@font}{\conditionally@traceoff}}\%
\@g@addto@macro{\pickup@font}{\escapechar\m@ne\langle\package\rangle.\global\MT@innannotttrue
\@g@addto@macro{\pickup@font}{\debug \MT@glet\MT@pdf@annot\@empty\debug \MT@glet{(line \number\inputlineno)}\langle\package\rangle.\fi

If \MT@font is empty, no substitution has taken place, hence \font@name is correct. Otherwise, if they are different, \font@name does not describe the font actually used. This test will catch first order substitutions, like bx to b, but it will still fail if the substituting font is itself substituted.
\MT@let@cn{\MT@font}{MT@subst@\expandafter\string}{\font@name}\%
\ifx\MT@font\relax\let\MT@font{\font@name}\else
\ifx\MT@font{\font@name}\else

\pickup@font
Remember the patched command, because we may have to disable ourselves in certain situations.

Additionally, we hook into \do@subst@correction, which is called if a substitution has taken place, to record the name of the ersatz font. Unfortunately, this will only work for one-level substitutions. We have to remember the substitute for the rest of the document, not just for the first time it is called, since we need it every time a font is letterspaced.

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.

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\do@subst@correction
\add@accent

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.
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.

\MT@register@font@cx

Add the substituted font to each feature list.

\MT@register@font@cx

For each feature, add the current font to the list, unless we didn’t set it up.

\MT@maybe@rem@from@list

Recursive through all context font lists of the document and remove the font, unless it’s the current context.

\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.

We need to ensure that math fonts are set up anew.
\textmicrotypecontext This is just a wrapper around \microtypecontext.

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

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

\MT@reset@context We must also keep track of all contexts in the document.

The next time we see the font, we have to reset all factors.
We also allow the `activate` shortcut.

```latex
\define@key{MTC}{activate}
{}{%
\setkeys{MT}{protrusion={#1}}%
\setkeys{MT}{expansion={#1}}%
}
```

```
\MT@pr@context
\MT@ex@context
\MT@tr@context
\MT@sp@context
\MT@kn@context
```

Initialise the contexts.

```
\MT@exp@one@n\MT@map@clist@n\MT@features,nl{%
\MT@def@n{MT@#1@context}{@}%
\MT@def@n{MT@#1@doc@contexts}{{@}}%
}
```

```
\let\MT@extra@context\@empty
```

```
14.3 Configuration
```

```
14.3.1 Font sets
```

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

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

```latex
\def\DeclareMicrotypeSet{%
\MT@begin@catcodes
\@ifstar
\MT@DeclareSetAndUseIt
\MT@DeclareSet
}\end@catcodes
```

We need to remember the name of the set currently being declared.

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

Define the current set name and parse the keys.

```latex
\def\MT@declare@sets#1#2#3{%
\def\MT@curr@set@name{#2}%
\MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
\MT@warning{Redefining \nameuse{MT@abbr@#1} set \MT@curr@set@name'}%
\MT@exp@one@n\MT@declare@sets
}\csname MT@rbba@##1\endcsname{#2}{#3}%
}
```
We do not add the expanded value to the list ... 
   but keep in mind that the list has to be expanded at the end of the preamble.
   
   We do not add the expanded value to the list ... 
   ... but keep in mind that the list has to be expanded at the end of the preamble.

Saying, for instance, ‘family=rms’ or ‘shape=bf’ will expand to \rmdefault resp. \bfdefault.

And ‘family = *’ will become \familydefault.

Test whether the command is actually defined.

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

It the last character is an asterisk, execute the second argument, otherwise the first one.

Fully expand the font specification and fix catcodes for all font sets. Also remove fontspec's counters.
Font sizes may also be specified as ranges. This has been requested by Andreas Bühmann, who has also offered valuable help in implementing this. Now, it is for instance possible to set up different lists for fonts with optical sizes. (The MinionPro project does this for the OpenType version of Adobe’s Minion. (Available from CTAN at pkg/minionpro))

Ranges will be stored as triplets of \{lower bound\} \{upper bound\} \{list name\}. For simple sizes, the upper boundary is −1.

2048 pt is \TeX’s maximum font size.
\MT@get@size \textbf{Translate a size selection command and normalise it.}

\MT@get@size \ def\MT@get@size{\
  \if*\MT@val\relax \\
  \def\@tempa\{\normalsize}\\
  \else \\
  \MT@let@cn\@tempa\{\MT@val}\\
  \fi \\
  \ifx\@tempa\relax \else \\
  The \texttt{relsize} solution of parsing \texttt{@setfontsize} does not work with the AMS classes, among others. I hope my hijacking doesn't do any harm. We redefine \texttt{@setfontsize} instead of \texttt{@setfontsize} because some classes might define the size selection commands by simply using \texttt{\fontsize} (e.g., the \texttt{a0poster} class). \\
  \begingroup \\
  \def\set@fontsize##1##2##3##4\@nil\{\endgroup\def\MT@val{##2}}\\
  \@tempa\@nil \\
  \fi \\
  \fi \\
  Test whether we finally got a number or dimension so that we can strip the \texttt{'}pt\texttt{'} (\texttt{@defaultunits} and \texttt{@strip@pt} are kernel macros). \\
  \MT@ifdimen\MT@val\relax \\
  \@defaultunits\@tempdima\MT@val pt\relax\@nnil \\
  \edef\MT@val{\strip@pt\@tempdima} \\
  \else \\
  \MT@warning{Could not parse font size \texttt{\MT@val}'\MessageBreak \\
  in font set \texttt{\MT@curr@set@name}} \\
  \let\MT@val\relax \\
  \fi \\
}

\MT@define@set@key@font \MT@define@set@key@font#1{\
  \define@key{MT@#1@set}{font}[{}]{
    \MT@glet@nc{MT@#1list@font@\MT@curr@set@name}\@empty \\
    \MT@map@clist@n{##1}{
      \def\MT@val{####1}\\
      \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax \\
      \expandafter\MT@get@font\MT@val\\*/\@nil \\
      \MT@exp@two@n\g@addto@macro \\
      \csname MT@#1list@font@\MT@curr@set@name\expandafter\endcsname \\
      {\MT@val,} \\
    } \\
    \expandafter\g@addto@macro\expandafter\MT@font@sets \\
    \csname MT@#1list@font@\MT@curr@set@name\endcsname \\
  } \\
}

\MT@get@font \MT@get@font \ def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{\
  \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0} \\
}
We can finally assemble all pieces to define \DeclareMicrotypeSet’s keys. They are also used for \DisableLigatures.

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.
IMPLEMENTATION: Configuration

Only use sets that have been declared.

\\MT@pr@setname
\\MT@ex@setname
\\MT@tr@setname
\\MT@sp@setname
\\MT@kn@setname
\\MT@use@set
\\MT@default@pr@set
\\MT@default@ex@set
\\MT@default@tr@set
\\MT@default@sp@set
\\MT@default@kn@set
\\MT@set@default@set

\MT@set@default@set
\def\MT@set@default@set#1#2{%
\MT@ifdefined@n@TF{MT@#1@set@@#2}{%
〈debug〉\MT@dinfo{1}{declaring default \@nameuse{MT@abbr@#1} set `#2'}%
\MT@xdef@n{MT@default@#1@set}{#2}%
}{%
\MT@error{%
The \@nameuse{MT@abbr@#1} set `#2' is undeclared. Using set \@nameuse{MT@#1@setname} instead}{}%
\MT@xdef@n{MT@default@#1@set}{all}%
}%
}%
}

\DeclareMicrotypeSetDefault
\def\DeclareMicrotypeSetDefault{%
\MT@begin@catcodes%
\MT@DeclareMicrotypeSetDefault%
}\MT@DeclareMicrotypeSetDefault
\newcommand\MT@DeclareMicrotypeSetDefault[2][]{%
\MT@ifempty{#1}{%
\MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
}{%
\MT@map@clist@n{#1}{{%
\MT@ifempty{##1}{}{%
\MT@is@feature{##1}{declaration of default set `#2'}{%\MT@set@default@set{\csname MT@rbba@##1\endcsname}{#2}%;}{%
}%
}%
\MT@error{%
The \@nameuse{MT@abbr@#1} set `#2' is undeclared. Using set \@nameuse{MT@#1@setname} instead}{}%
}%
}}%
\MT@end@catcodes%
}

\MT@default@pr@set
\MT@default@ex@set
\MT@default@tr@set
\MT@default@sp@set
\MT@default@kn@set
\MT@set@default@set
\def\MT@set@default@set#1#2{%
\MT@ifdefined@n@TF{MT@#1@set@@#2}{%
\MT@map@clist@n{#1}{{%
\MT@ifempty{##1}{}{%
\MT@is@feature{##1}{activation of set `#2'}{%\MT@exp@one@n\MT@use@set{\csname MT@rbba@##1\endcsname}{#2};}{%
}%
}%
}}%
\MT@end@catcodes%
}

\MT@pr@setname
\MT@ex@setname
\MT@tr@setname
\MT@sp@setname
\MT@kn@setname
\MT@use@set

\MT@default@pr@set
\MT@default@ex@set
\MT@default@tr@set
\MT@default@sp@set
\MT@default@kn@set
\MT@set@default@set
\def\MT@set@default@set#1#2{%
\MT@ifdefined@n@TF{MT@#1@set@@#2}{%
\MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
}{%
\MT@error{%
The \@nameuse{MT@abbr@#1} set `#2' is undeclared. Using set \@nameuse{MT@#1@setname} instead}{}%
\MT@xdef@n{MT@default@#1@set}{all}%
}%
}%

\DeclareMicrotypeSetDefault
\def\DeclareMicrotypeSetDefault{%
\MT@begin@catcodes%
\MT@DeclareMicrotypeSetDefault%
}\MT@DeclareMicrotypeSetDefault
\newcommand\MT@DeclareMicrotypeSetDefault[2][]{%
\MT@ifempty{#1}{%
\MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
}{%
\MT@map@clist@n{#1}{{%
\MT@ifempty{##1}{}{%
\MT@is@feature{##1}{declaration of default set `#2'}{%\MT@set@default@set{\csname MT@rbba@##1\endcsname}{#2}%;}{%
}%
}%
\MT@error{%
The \@nameuse{MT@abbr@#1} set `#2' is undeclared. Using set \@nameuse{MT@#1@setname} instead}{}%
}%
}}%
\MT@end@catcodes%
}
14.3.2 Variants and aliases

\DeclareMicrotypeVariants Specify suffixes for variants (see fontname/variants.map). The starred version appends to the list.

\let\MT@variants\@empty
\def\DeclareMicrotypeVariants{% 
  \MT@begin@catcodes 
  \@ifstar\MT@DeclareVariants{\let\MT@variants\@empty\MT@DeclareVariants}% 
  \MT@end@catcodes 
}\MT@DeclareVariants

\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.

\DeclareMicrotypeAlias
\newcommand*\MT@DeclareMicrotypeAlias[2]{% 
  \def\@tempb{#2}% 
  \@onelevel@sanitize\@tempb 
  \MT@ifdefined@n@T{MT@#1@alias}{% 
    \MT@warning{Alias font family \@tempb will override alias \@nameuse{MT@#1@alias}' for font family `#1'}}% 
  \MT@xdef@n{MT@#1@alias}{\@tempb}% 
}\MT@DeclareMicrotypeAlias

\LoadMicrotypeFile May be used to load a configuration file manually.

\LoadMicrotypeFile


14.3.3 Disabling ligatures

\DisableLigatures \MTIDisableLigatures \MTOnl@setname

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

The optional argument may be used to disable selected ligatures only.

\MTOnl@ligatures

If pdfTeX is too old, we throw an error.

14.3.4 Interaction with babel

\DeclareMicrotypeBabelHook

Declare the context that should be loaded when a babel language is selected. The command will not check whether a previous declaration will be overwritten.

14.3.5 Fine tuning

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

This macro accepts three arguments: [options,] set of font attributes and list of character protrusion factors.

A new macro called \MT@pr@c@ will be defined to be \textlangle #3 \rangle (i.e., the list of characters, not expanded).

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

\SetTracking
We first set the name (if specified), then remove it from the list, and set the remaining keys.
\edef\x##1name=##2,##3\@nil{\setkeys{#1}{name=##2}\gdef\MT@options{##1##3}\MT@rem@from@clist{name=}\MT@options}\x#2,name=,\@nil\@expandtwoargs\setkeys{#1}\MT@options}

\MT@define@code@key Define the keys for the configuration lists (which are setting the codes, in pdf\TeX speak).
\def\MT@define@code@key#1#2{\define@key{MT@#2}{#1}[]}\@tempcnta=\@ne\MT@map@clist@n{##1}{\KV@@sp@def\MT@val{####1}\MT@get@highlevel{#1}\MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}\advance\@tempcnta\@ne}\MT@define@code@key@family
\MT@define@code@key@size \MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.
\MT@define@code@key@font \MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.

\MT@define@code@key@family Remove fontspec's internal feature counter.
\def\MT@define@code@key@family#1{\define@key{MT@#1}{family}[]}\@tempcnta=\@ne\MT@map@clist@n{##1}{\KV@@sp@def\MT@val{####1}\MT@get@highlevel{family}\if\MT@fontspec\edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x\fi\MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}\advance\@tempcnta\@ne}\MT@define@code@key@size
\MT@define@code@key@font

Here, too, we allow for something like ‘bf\^*’. It will be expanded immediately.
\MT@define@code@key@size
\MT@define@code@key@font
\MT@get@font@and@size\ translate any asterisks and split off the size.
\MT@define@opt@key\ The options in the optional first argument.
\MT@listname@count\ The options in the optional first argument.
\MT@define@opt@key\ Only one context is allowed. This might change in the future.
\MT@define@opt@key\ Automatically enable font copying if we find a protrusion or expansion context.
After the preamble, check whether font copying is enabled. For older pdftex versions, disallow. It also works with LuaTeX 0.30 or newer.

Protrusion contexts might also work without copying the font, so we don’t issue an error but only a warning. The problem is that pdftex only allows one set of protrusion factors for a given font within one paragraph (those that are in effect at the end of the paragraph will be in effect for the whole paragraph). When different fonts are loaded – like in the example with the footnote markers – we don’t need to copy the fonts.
Protrusion codes may be relative to character width, or to any dimension.

Tracking may only be relative to a dimension.

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

The first argument to \SetExpansion accepts some more options.
Don’t use autoexpand for pdfTeX version older than 1.20.

Tracking: Interword spacing and outer kerning. The variant with space just in case \SetTracking is called inside an argument (e.g., to \IfFileExists).

Which ligatures should be disabled?

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.

The optional argument may be used to restrict the list to some features, and to specify an input encoding.

14.3.6 Character inheritance

\DeclareCharacterInheritance

\MT@inh@feat \MT@extra@inputenc
\let\MT@inh@feat\@empty
\setkeys{MT@inh@}{#1}%%
\MT@begin@catcodes
\MT@set@inh@list
Safe category codes.\MT@set@inh@list
\def\MT@set@inh@list#1#2{%
\ifempty\MT@inh@feat{%
\MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
} {%
\MT@map@clist@c\MT@inh@feat{{%
\KV@@sp@def\@tempa{##1}%
\ifempty\@tempa\relax{%
\MT@exp@one@n\MT@declare@char@inh
{\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
}%
}%
}%
\MT@end@catcodes
}
The keys for the optional argument.\MT@map@clist@c\MT@features@long{%
\define@key{MT@inh@}{#1}{[]}{{\edef\MT@inh@feat{\MT@inh@feat#1,}}}%
\define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
\MT@declare@char@inh
The lists cannot be given a name by the user.\MT@define@code@key{encoding}{inh}%
\MT@define@code@key@family{inh}%
\MT@define@code@key{series}{inh}%
\MT@define@code@key@size{inh}%
\MT@define@code@key@font{inh}%
\MT@inh@do
Now parse the third argument, the inheritance lists. We define the commands \MT@inh@{\(name\)}@{\(slot\)}, containing the inheriting characters. They will also be translated to slot numbers here, to save some time. The following will be executed only once, namely the first time this inheritance list is encountered (in \MT@set@{\(feature\)}@codes).
\MT@inh@do
Parse the second argument. \DeclareCharacterInheritance may also be set up for various combinations. We can reuse the key setup from the configuration lists (\Set...).
Only gather the inheriting characters here. Their codes will actually be set in `\MT@set@{feature}@codes`.

3772 \MT@inh@split
3773 \MT@set@{feature}@codes
3774 \MT@get@slot
3775 \MT@get@slot
3776 \MT@get@slot
3777 \MT@get@slot
3778 \MT@get@slot
3779 \MT@get@slot
3780 \MT@get@slot
3781 \MT@get@slot
3782 \MT@get@slot
3783 \MT@get@slot
3784 \MT@get@slot
3785 \MT@get@slot
3786 \MT@get@slot
3787 \MT@get@slot
3788 \MT@get@slot
3789 \MT@get@slot
3790 \MT@get@slot
3791 \MT@get@slot
3792 \MT@get@slot
3793 \MT@get@slot
3794 \MT@get@slot
3795 \MT@get@slot
3796 \MT@get@slot

14.3.7 Permutation

Calling `\MT@permute` will define commands for all permutations of the specified font attributes of the form `\MT@{list type}/@encoding/@family/@series/@shape/\{\}` to be the expansion of `\MT@{list type}@name`, i.e., the name of the currently defined list. Size ranges are held in a separate macro called `\MT@{list type}/font axes@sizes`, which in turn contains the respective `list name`\s attached to the ranges.

3797 \MT@permute
3798 \MT@permute@
3799 \MT@permute@@
3800 \MT@permute@@@@
3801 \MT@permute@@@
In order to save some memory, we can ignore unused encodings (inside the document).

Some sanity checks: an encoding must be specified (unless nothing else is).

Add the list of ranges to the beginning of the current combination, after checking for conflicts.
\MT@warning{\@nameuse{MT@abbr@MT@permutelist} list `\@nameuse{MT@MT@permutelist @name}' will override list `\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'

for `\MessageBreak font `\@tempa'}%
\% \%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
* Item in list is a simple size, new item is a range.
\MT@ifdim\@tempb>{#1}\relax{% 
\MT@ifdim\@tempc>{#1}{% 
\MT@if@true 
\edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}% 
\relax 
}\relax 
}% 
}% 
}\relax 
}% 
\MT@ifdim\@tempc=\m@ne{% 
* Item in list is a range, new item is a simple size.
\MT@ifdim\@tempb<{#2}{% 
\MT@ifdim\@tempb<{#1}\relax\MT@if@true 
}\relax 
}% 
\MT@ifdim\@tempb<{#2}{% 
\MT@ifdim\@tempc>{#1}{% 
\MT@if@true 
\edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}% 
\relax 
}\relax 
}% 
\ifMT@if@ 
\MT@ifstreq{#3}{}{\csname MT@permutelist @\csname MT@permutelist @name\endcsname @load\endcsname}{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}% 
\MT@optwarn@admissible
\def\MT@optwarn@admissible#1#2{% 
\MT@warning@nl{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}\MT@optwarn@nan
\ifMT@opt@expansion
\fi 
\MT@opt@auto
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}\MT@optwarn@nan
\MT@optwarn@nan
\ifMT@opt@expansion 
\MT@opt@auto
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}\MT@optwarn@nan
\MT@optwarn@nan
\ifMT@opt@expansion 
\MT@opt@auto
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}\MT@optwarn@nan
\MT@optwarn@nan
\ifMT@opt@expansion 
\MT@opt@auto
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}14.4 Package options

14.4.1 Declaring the options
\ifMT@opt@expansion 
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}\MT@optwarn@nan
\MT@optwarn@nan
\ifMT@opt@expansion 
\MT@opt@auto
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}\MT@optwarn@nan
\MT@optwarn@nan
\ifMT@opt@expansion 
\MT@opt@auto
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
\edef\MT@optwarn@admissible#1#2{% 
\MT@warning{`#1' is not an admissible value for option `#2'. Assuming `false'}% 
}
\MT@warning{Value `#1' for option `#2' is not a number.}
Using default value of \number{\@nameuse{MT@#2@default}}
\MT@opt@def@set
\def\MT@opt@def@set#1{% 
\MT@ifdefined@TF{MT@#1@set@@\MT@val}{% 
\MT@xdef{n}{MT@#1@setname}{\MT@val}% 
}% 
\MT@warning{The set `\MT@val' is undeclared. Using set `\@nameuse{MT@#1@setname}' instead}% }

expansion and protrusion may be true, false, compatibility,
and/or a set name.
\MT@map@clist{n}{protrusion,expansion}{% 
\define@key{MT}{#1}[true]{% 
\csname MT@opt@#1true\endcsname 
\MT@map@clist{n}{##1}{% 
\KV@@sp@def\MT@val{####1}% 
\MT@ifempty\MT@val\relax{% 
\csname MT@#1true\endcsname 
\edef\@tempb{\csname MT@rbba@#1\endcsname}% 
\MT@ifstreq\MT@val{true}\relax{% 
\MT@ifstreq\MT@val{false}\relax{% 
\MT@ifstreq\MT@val{compatibility}\relax{% 
\MT@let{MT@#1level}{1}% 
}% 
\MT@ifstreq\MT@val{nocompatibility}\relax{% 
\MT@let{MT@#1level}{0}% 
}% 
}% 
}% 
\MT@opt@def@set{#1}%
}%

If everything failed, it should be a set name.
\MT@opt@def@set{#1}%

activate is a shortcut for protrusion and expansion.
\define@key{MT}{activate}[true]{% 
\setkeys{MT}{protrusion={#1}}% 
\setkeys{MT}{expansion={#1}}% }

spacing, kerning and tracking do not have a compatibility level.
\MT@map@clist{n}{spacing,kerning,tracking}{% 
\define@key{MT}{#1}[true]{% 
\MT@map@clist{n}{##1}{% 
\KV@@sp@def\MT@val{####1}% 
\MT@ifempty\MT@val\relax{% 
\csname MT@#1true\endcsname 
\edef\@tempb{\csname MT@rbba@#1\endcsname}% 
\MT@ifstreq\MT@val{true}\relax{% 
\MT@ifstreq\MT@val{false}\relax{% 
\MT@ifstreq\MT@val{compatibility}\relax{% 
\MT@let{MT@#1level}{1}% 
}% 
\MT@ifstreq\MT@val{nocompatibility}\relax{% 
\MT@let{MT@#1level}{0}% 
}% 
}% 
}% 
}% 
}%

activate is a shortcut for protrusion and expansion.
\define@key{MT}{activate}[true]{% 
\setkeys{MT}{protrusion={#1}}% 
\setkeys{MT}{expansion={#1}}% }

spacing, kerning and tracking do not have a compatibility level.
The true/false options: draft, final (may be inherited from the class options), auto, selected, babel, DVIoutput, defersetup, copyfonts.

Boolean options that only set the switch.

Setting the defersetup option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.
copyfonts will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This option is also *undocumented* in the hope that we can always find out automatically whether it’s required. It also works with LuaTeX 0.30 or newer.

For verbose output, we redefine \texttt{MT\@vinfo}.

Take problems seriously.

Cast warnings to the winds.
Options with numerical keys: factor, stretch, shrink, step, letterspace.  

No nonsense in \MT@factor et al.? A space terminates the number.  

factor will define the protrusion factor only.  

Unit for protrusion codes.  

\begin{verbatim}
14.4.2 Loading the definition file
\end{verbatim}
14.4.3 Reading the configuration file

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as modern \TeX systems have switched to the pdf\TeX engine even for DVI output, so that the user might not even be aware of the fact that she's running pdf\TeX.)

\MT@protrusiontrue
\MT@protrusiontrue
\package{pdftex-def}\latex-def
\ifnum\pdfoutput<\@ne \else
\MT@requires@pdftex4{
\MT@expansiontrue
\MT@autotrue
}\relax
\fi
\MT@autotrue
\MT@autotrue
\PDFTEX\LATEX
\MT@check@active@set

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.

\MT@config@file
\MT@get@config
\MT@check@active@set

Load the file.
\IfFileExists{\MT@config@file}{%
\MT@begin@catcodes
\let\MT@begin@catcodes\relax
\MT@end@catcodes
\MT@curr@file\MT@config@file
\input{\MT@config@file}
\endgroup
}{%\MT@warning@nl{%
\MT@warning@nl{Could not find configuration file `\MT@config@file'!\MessageBreak
This will almost certainly cause undesired results.\MessageBreak
Please fix your installation}%
}%

\MT@check@active@set

We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by \texttt{\DeclareMicrotypeSetDefault} (this is done at the end of the preamble).
\MT@check@active@set
If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set \empty', and issue a warning.

\MT@define@optionX\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 \ifpackage{loaded} in the font package was not viable), and (2) checking \AtBeginDocument could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren't overwritten. Example:

\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\ifpackageloaded{microtype}
\MinionPro@MT@Hook
\else
\ifundefined{Microtype@Hook}
\let\Microtype@Hook\MinionPro@MT@Hook
\else
\g@addto@macro\Microtype@Hook{\MinionPro@MT@Hook}\fi\fi

\MicroType@Hook with a capital T (which only existed in version 1.7) is provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

\MT@ifdefined@c@T\Microtype@Hook{\MT@warning{Command \string\Microtype@Hook\space is deprecated. Use \string\Microtype@Hook\space instead}\Microtype@Hook}
\MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook

14.4.5 Changing options later

Inside the preamble, \microtypesetup accepts the same options as the package (unless deferersetup=false). In the document body, it accepts the options: protrusion, expansion, activate, tracking, spacing and kerning. Specifying font sets is not allowed.
Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

\MT@checksetup[#1]{%
\@tempcnta=\csname MT@\@tempb @level\endcsname
\MT@vinfo{Enabling #1 (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
}%
\MT@ifstreq\MT@val{true}{%
\MT@checksetup{#1}{%
\@tempcnta=\csname MT@\@tempb @level\endcsname
\MT@vinfo{Enabling #1 (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
}%
}\MT@ifstreq\MT@val{false}{%
\@tempcnta=\z@
\MT@vinfo{Disabling #1\on@line}%
}}{%
\MT@ifstreq\MT@val{compatibility}{%
\MT@checksetup{#1}{%
\@tempcnta=\one
\MT@let@nc{MT@\@tempb @level}\one
\MT@vinfo{Setting #1 to level 1\on@line}%
}\MT@ifstreq\MT@val{nocompatibility}{%
\MT@checksetup{#1}{%
\@tempcnta=\two
\MT@let@nc{MT@\@tempb @level}\two
\MT@vinfo{Setting #1 to level 2\on@line}%
}\MT@error{Value `\MT@val' for key `#1' not recognised}
}{Use any of `true', `false', `compatibility' or `nocompatibility'.}%
}%
}%
\ifnum\@tempcnta>\m@ne
#2\@tempcnta\relax
\fi
}%
}
\MT@checksetup{Test whether the feature wasn’t disabled in the package options.

\def\MT@checksetup#1{%
\csname ifMT@#1\endcsname
\expandafter\@firstofone
\else
\MT@error{You cannot enable #1 if it was disabled\on@line}
\fi
\expandafter\@gobble
}
\MT@define@optionX{protrusion}{\MT@protrudechars}
\MT@define@optionX{expansion}{\MT@adjustspacing}
\MT@requires@luatex{\let\pdfprotrudechars\protrudechars
\let\pdfadjustspacing\adjustspacing}
\MT@protrudechars
\MT@adjustspacing
IMPLEMENTATION: Package options

\MT@define@optionX@

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

\MT@define@optionX@

We cannot simply let \MT@tracking relax, since this may select the already letter-spaced font instance.

\MT@define@optionX@

Disable for older pdfTeX versions and for Xe\TeX{} and Lua\TeX. 
Disable everything – may be used as a temporary work-around in case setting up fonts doesn’t work under certain circumstances, but only until that specific problem is fixed. This is \textit{undocumented}, as it completely deprives us of the possibility to act – we’re blind and paralysed.

\begin{verbatim}
\MT@saved@setupfont
\let\MT@saved@setupfont\MT@setupfont
\define@key{MTX}{disable}[]{\MT@info{Inactivate `\MT@MT' package}\\\let\MT@setupfont\relax}
\define@key{MTX}{enable}[]{\MT@info{Reactivate `\MT@MT' package}\\\let\MT@setupfont\MT@saved@setupfont}
\end{verbatim}

\subsection{Processing the options}

\begin{verbatim}
\MT@ProcessOptionsWithKV
\end{verbatim}

\begin{verbatim}
\MT@Getkey
\end{verbatim}

Now we can take the appropriate actions. We also tell the log file which options the user has chosen (in case it’s interested).

\begin{verbatim}
\MT@Getkey{\MT@warning@nl{`draft' option active.\\Disabling all micro-typographic extensions.}}
\end{verbatim}
This might lead to different line and page breaks.}
\let\MT@setupfont\relax
\renewcommand*\LoadMicrotypeFile[1]{%}
\renewcommand*\microtypesetup[1]{%}
\renewcommand*\microtypecontext[1]{%}
\renewcommand*\lsstyle{}%
\else
\MT@setup@PDF
\MT@setup@copies

\MT@setup@PDF

Fix the font sets.
\MT@map@tlist@c\MT@font@sets\MT@fix@font@set
\MT@setup@protrusion
\MT@setup@expansion
\MT@setup@tracking
\MT@setup@warntracking
\MT@setup@spacing
\MT@setup@kerning
\MT@setup@noligatures
\}
\MT@setup@PDF
\MT@setup@copies

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 \pdfoptions 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@setup@copies

Working on font copies?
\MT@setup@copies
\exetext-def
\MT@setup@PDF
\MT@setup@copies

Protrusion.
\MT@setup@protrusion

\MT@setup@protrusion

\MT@setup@expansion  For DVI output, the user must have explicitly passed the expansion option to the package.

\begin{verbatim}
\ifnum\pdfoutput<\@ne
\ifMT@opt@expansion \else \MT@expansionfalse \fi
\fi
\ifMT@expansion
Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.
\ifnum\MT@stretch=\@ne
\let\MT@stretch\MT@stretch@default \fi
\ifnum\MT@shrink=\@ne
\let\MT@shrink\MT@stretch \fi
\ifnum\MT@step=\@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
\edef\MT@step{\number\@tempcnta}\fi
\ifnum\MT@step=\z@
\MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak Setting it to one}\def\MT@step{1}\fi
\fi
\def\MT@auto
Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the \texttt{hz} programme really usable. It must be either \texttt{autoexpand} or empty (or \texttt{1000} for older versions of pdfTeX). With LuaTeX, we just leave it empty, as there's actually no difference – non-automatic font expansion doesn't work anymore. In LuaTeX 1.0.6, the \texttt{autoexpand} option seems to have been removed altogether and would
\end{verbatim}

We turn off automatic expansion if output mode is DVI and we're running pdfTeX.

Also, if pdfTeX is too old.

No automatic expansion.

Choose the appropriate macro for selected expansion.

Filter out stretch=0, shrink=0, since it would result in a pdfTeX error.
\MT@check@step  \ Check whether stretch and shrink are multiples of step.
\def\MT@check@step##1{\@tempcnta=\csname MT@##1\endcsname\divide\@tempcnta \MT@step\multiply\@tempcnta \MT@step\ifnum\@tempcnta=\csname MT@##1\endcsname\else\MT@warning@nl{The ##1 amount is not a multiple of step.\MessageBreak\The effective maximum ##1 is \the\@tempcnta \space(step \number\MT@step)}\fi\fi\MT@check@step{stretch}\MT@check@step{shrink}\MT@check@active@set{ex}\Inside \showhyphens, font expansion should be disabled. (Since 2017/01/10, the \TeX format contains a different version for \LaTeX, but since expansion doesn’t work with \LaTeX, we don’t have to bother.)
\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 \texttt{ltfssbas.dtx})?
\def\showhyphens##1{\setbox0\vbox{\color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip\hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}\else\let\MT@expansion\relax\MT@info@nl{No font expansion}\fi\MT@setup@tracking  \ Tracking, spacing and kerning.
\ifpdf\checkcommand*{\showhyphens}{\setbox0\vbox{\color@begingroup\everypar{}\parfillskip\z@skip\hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}\else\MT@info@nl{No font expansion}\fi
\MT@setup@tracking
Enable protrusion for compensation at the line edges.

\ifMT@protrusion\else\MT@protrudechars\fi
\else
\let\MT@tracking\relax
\MT@info@nl{No adjustment of tracking}\
\fi

The \ragged2e package sets interword spaces to a fixed value without glue. \microtype’s modifications can therefore have undesired effects. Therefore, we issue a warning.

\ifMT@spacing
\edef\MT@active@features{\MT@active@features,sp}\
\pdfadjustinterwordglue\@ne
\MT@info@nl{Adjustment of interword spacing enabled}\
\else
\let\MT@spacing\relax
\MT@info@nl{No adjustment of interword spacing}\
\fi

Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue>0. Why 1500? Because some packages redefine \frenchspacing.\footnote{Cf. the c.t.t. thread ‘\frenchspacing with AMS packages and babel’, started by Philipp Lehman on 16 August 2005, MID: ddtbaj$rob$1@online.de}
If pdiTEx is too old, we disable tracking, spacing and kerning, and throw an error message. We also switch the features off for LuaTeX and XeTeX.

\MT@error@doesnt@work
\MT@setup@tracking
\MT@setup@kerning
\MT@setup@spacing

With pdiTEx, we issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn’t already, to compensate for the letterspacing kerns.
\MT@setup@moligatures
\DisableLigatures is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.
\MT@addto@setup{\MT@documenttrue}
\MT@shorthandoff
Active characters can only be switched off if babel isn’t loaded after microtype.
\MT@addto@setup{\ifMT@babel
\MT@shorthandoff#1#2{\MT@error{You must load `babel' before `\MT@MT'
Otherwise, `\MT@MT' cannot switch off #1 babel’s active characters.}}
\MT@addto@setup{\ifMT@babel
We patch the language switching commands to enable language-dependent setup.
\@ifpackageloaded{babel}{%
\MT@info@nl{Redefining babel's language switching commands}%
\let\MT@orig@select@language\select@language
\def\select@language#1{%\MT@orig@select@language{#1}%\MT@set@babel@context{#1}%}
\let\MT@orig@foreign@language\foreign@language
\def\foreign@language#1{%\MT@orig@foreign@language{#1}%\MT@set@babel@context{#1}%}
}\ifMT@kerning
\MT@if@false\MT@with@babel@and@T{french}\MT@if@true\MT@with@babel@and@T{frenchb}\MT@if@true\MT@with@babel@and@T{francais}\MT@if@true\MT@with@babel@and@T{canadien}\MT@if@true\MT@with@babel@and@T{acadian}\MT@if@true\ifMT@if@MT@shorthandoff{French}{:;!?}\fi
\MT@if@false\MT@with@babel@and@T{turkish}\MT@if@true\ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi\fi
\MT@set@babel@context\languagename}
\MT@ifdefined@c@T\MT@setup@spacing@check\MT@set@babel@context\languagename\MT@if@false\MT@warning@nl{You did not load the babel package.\MessageBreakThe `babel' option won't have any effect}%\fi
\MT@if@false\MT@addto@setup{\fi}
\selectfont}
\MT@curr@file
This is the current file (hopefully with the correct extension).
\edef\MT@curr@file{\jobname.tex}
\MT@requires@latex1{
\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@@empty}
\MT@restore@catcodes
That was that.
}
15 Configuration files

Let’s now write the font configuration files.

15.1 Font sets

We first declare some sets in the main configuration file.

```latex
\DeclareMicrotypeSet{all}
\DeclareMicrotypeSet{allmath}
\DeclareMicrotypeSet{alltext}
\DeclareMicrotypeSet{allmath-nott}
\DeclareMicrotypeSet{alltext-nott}
\DeclareMicrotypeSet{basicmath}
\DeclareMicrotypeSet{basictext}
\DeclareMicrotypeSet{smallcaps}
\DeclareMicrotypeSet{footnotesize}
\DeclareMicrotypeSet{scriptsize}
```
size = {-footnotesize}

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

The default sets.

\%\% DEFAULT SETS

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

\section{Font variants and aliases}

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

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

Other candidates: 2 (proportional digits), e (engraved), f (Fraktur), g (small text), h (shadow), l (outline), n (informal), p (ornaments), r (roman), s (sans serif), t (typewriter). I’ve omitted them since they seem hardly be used and/or they are actually more than just a variant, i.e., they shouldn’t share a file.

Fonts that are ‘the same’: The fontspec package will set \texttt{lmr} as the default font, whose declarations for EU1/EU2/TU encoding are in \texttt{mt-LatinModernRoman.cfg}. Since 2016/12/03, the default encoding with \TeX{} and \LaTeX{} in the \TeX{} format is TU, even if fontspec is not loaded.

\MT@if@false
\ifx\UnicodeEncodingName\@undefined\else
\MT@ifstreq{\encodingdefault}{\UnicodeEncodingName}\MT@if@true\relax
\fi
\ifMT@fontspec\MT@if@true\fi
\ifMT@if@
\DeclareMicrotypeAlias{lmr}{Latin Modern Roman}
\else
\DeclareMicrotypeAlias{lmr}{cmr} % lmodern
\fi

The Latin Modern fonts, the virtual fonts from the \texttt{ae} and \texttt{zefonts}, and the \texttt{eco} and \texttt{hfoldsty} packages (oldstyle numerals) all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later. We mustn’t forget the Latin Modern math fonts.

\begin{verbatim}
\DeclareMicrotypeAlias{lmsy}{cmsgy}
\end{verbatim}

The packages \texttt{pxfonts} and \texttt{txfonts} fonts inherit Palatino and Times settings respectively, also the \TeX{} Gyre fonts Pagella and Termes (formerly: \texttt{qfonts}).
The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

The \texttt{FPL} \texttt{Neu} fonts, a ‘re-implementation’ of Palatino.

The \texttt{newpx} package, a replacement for \texttt{pxfonts}.

The \texttt{newtx} package, a replacement for \texttt{txfonts}.

The \texttt{tempora} package.

The OpenType versions:

More Times variants, to be checked: \texttt{pns}, \texttt{mns} (\texttt{TimesNewRomanPS}); \texttt{mnt} (\texttt{TimesNewRomanMT}, \texttt{TimesNRSevenMT}), \texttt{mtm} (\texttt{TimesSmallTextMT}); \texttt{pte} (\texttt{TimesEuropa}); \texttt{ptt} (\texttt{TimesTen}); \texttt{TimesEighteen}; \texttt{TimesModernEF}.

The \texttt{eulervm} package virtually extends the Euler fonts.

MicroPress’s Charter version (\texttt{chmath}).

The \texttt{XCharter} package extends the Charter fonts.

The \texttt{mathdesign} package provides math fonts matching Bitstream Charter and URW Garamond.

The \texttt{garamondx} package, an extension of URW Garamond, providing small caps and oldstyle figures.

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the configuration.

Euro symbol fonts, to save some files.
15.3 Interaction with babel

Contexts that are to be set when switching to a language.

15.4 Note on admissible characters

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

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

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

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

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

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

With \TeX\ or \LaTeX, in contrast, it is advisable to use the proper Unicode characters.

15.5 Character inheritance

First the lists of inheriting characters. We only declare those characters that are the same on both sides, i.e., not CE for O.

15.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 ('fi' ligature), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), \AE, \ae, \OE, \oe.

\DeclareCharacterInheritance
{ encoding = OT1 }
{ f = {011}, \% ff
i = {\i},
j = {\j},
o = {\o} }

15.5.2 T1

Candidates here: 028 ('fi'), 029 ('fl'), 030 ('ffi'), 031 ('ffl'), 156 ('IJ' ligature, since \LaTeX 2005/12/01 accessible as \IJ, 188 ('ij', 'ij'), \AE, \ae, \OE, \oe.

\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 = {\`D,\DH},
d = {\`d,\d d},
E = {\`E,\`E,\^E,\v E,\k E,\v E},
e = {\`e,\`e,\^e,\v e,\k e,\v e},
f = {027}, \% ff
G = \{\u G\},
g = \{\u g\},
I = \{\`I,\v I,\l I,\i I\},
i = \{\`i,\v i,\l i,\i I\},
j = \{\j\},
L = \{\`L,\v L,\l L\},
l = \{\`l,\v l,\l L\},
N = \{\`N,\~N,\v N\},
n = \{\`n,\~n,\v n\},
O = \{\`O,\v O,\l O,\i O,\k O,\r O\},
o = \{\`o,\v o,\l o,\i o,\k o,\r o\},
R = \{\`R,\v R\},
r = \{\`r,\v r\},
S = \{\`S,\c S,\v S,\SS\},
s = \{\`s,\c s,\v s\},
T = \{\`T,\v T\},
t = \{\`t,\v t\},
U = \{\`U,\v U,\l U,\i U,\k U,\r U\},
u = \{\`u,\v u,\l u,\i u,\k u,\r u\},
Y = \{\`Y,\v Y\},
y = \{\`y,\v y\},
Z = \{\`Z,\l Z,\v Z\},
z = \{\`z,\l z,\v z\}

The 'soft hyphen' often has reduced right side bearing so that it may already be protruded, hence no inheritance.

\% - = {127},
15.5.3 LY1

More characters: 008 (‘fl’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), AÆ, æ, Æ, œ.

\DefineCharacterInheritance
\{ encoding = LY1 \}
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’},
L = {‘\L’},
l = {‘\l’},
N = {‘\N’},
n = {‘\n’},
o = {‘\o’, ‘\o’, ‘\o’, ‘\o’, ‘\o’, ‘\o’},
S = {‘\S’},
s = {‘\s’},
U = {‘\U’, ‘\U’, ‘\U’, ‘\U’},
u = {‘\u’, ‘\u’, ‘\u’, ‘\u’},
Y = {‘\Y’, ‘\Y’},
y = {‘\y’, ‘\y’},
Z = {‘\Z’},
z = {‘\z’})

15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 (‘fk’), 012 (‘fi’),
013 (‘ff’), 014 (‘ffi’), 015 (‘ffl’), AÆ, æ, Æ, œ.

\DefineCharacterInheritance
\{ encoding = OT4 \}
( A = {‘k A’},
a = {‘k a’},
C = {‘\c’},
c = {‘\c’},
E = {‘k E’},
e = {‘k e’},
f = {011}, % ff
I = {‘\i’},
j = {‘\j’},
L = {‘\L’},
l = {‘\l’},
N = {‘\N’},
n = {‘\n’},
O = {‘\O’, ‘\O’},
o = {‘\o’, ‘\o’},
S = {‘\S’},
s = {‘\s’},
Z = {‘\Z’, ‘\Z’},
z = {‘\z’, ‘\z’},
\textquotedblleft = “FF
\textquoteright” = "FF

\)
15.5.5 QX

The Central European QX encoding.\textsuperscript{16} Ligatures: 009 (‘ł’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’).

\begin{verbatim}
\DeclareCharacterInheritance
  { encoding = QX }
  { A = {\`A,\'A,\^A,\~A,\"A,\l A,\AA},
    a = {\`a,\'a,\^a,\~a,\"a,\l a,\aa},
  C = {\`C,\c C},
  c = {\`c,\c c},
  D = {\XD},
  E = {\`E,\'E,\^E,\~E,\"E,\l E},
  e = {\`e,\'e,\^e,\~e,\"e,\l e},
  f = {011}, \% ff
  I = {\`I,\'I,\^I,\~I,\"I,\l I},
  i = {\`i,\'i,\^i,\~i,\"i,\i},
  j = {\j},
  L = {\L},
  l = {\l},
  N = {\`N,\~N},
  n = {\n,\~n},
  O = {\O,\`O,\'O,\^O,\~O,\"O},
  o = {\o,\`o,\'o,\^o,\~o,\"o},
  S = {\`S,\c S,\textcommabelow S,\v S},
  s = {\s,\c s,\textcommabelow s,\v s},
  T = {\c T,\textcommabelow T},
  t = {\t,\textcommabelow t},
  U = {\`u,\u,\^u,\~u,\"u,\l u},
  u = {\`u,\u,\^u,\~u,\"u,\l u},
  Y = {\`Y,\Y},
  y = {\y,\Y},
  Z = {\`z,\c z,\textcommabelow z,\v z},
  z = {\z,\c z,\textcommabelow z,\v z},
  = \textellipsis
}
\end{verbatim}

The Romanian \textcommabelow accents are actually replacements for the \c variants, which had previously (and erroneously\textsuperscript{17}) been included in QX encoding. They are still kept for backwards compatibility.

15.5.6 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented characters that there is no room for any ligatures.

\begin{verbatim}
\DeclareCharacterInheritance
  { encoding = T5 }
  { A = {\`A,\'A,\`h A,\d A,\^A,\~A,\u A,
    \Acircumflex,\Acircumflex,\textcommabelow Acircumflex,\v Acircumflex,\d Acircumflex,
    \Abreve,\Abreve,\textcommabelow Abreve,\h Abreve,\d Abreve,\l Abreve,\v Abreve,\d Abreve},
  a = {\`a,\'a,\`h a,\d a,\^a,\~a,\u a,
    \acircumflex,\acircumflex,\textcommabelow acircumflex,\v acircumflex,\d acircumflex,
    \abreve,\abreve,\textcommabelow abreve,\h abreve,\d abreve,\l abreve,\v abreve,\d abreve},
  D = {\DJ},
  d = {\dj},
  E = {\`E,\'E,\`h E,\d E,\^E,\v E,
    \Ecircumflex,\Ecircumflex,\textcommabelow Ecircumflex,\v Ecircumflex,\d Ecircumflex},
  e = {\`e,\e,\`h e,\d e,\e,\v e,
    \ecircumflex,\ecircumflex,\textcommabelow ecircumflex,\v ecircumflex,\d ecircumflex},
\end{verbatim}

\footnote{Contributed by Maciej Eder.}
\footnote{Cf. \url{http://tug.org/pipermail/tex-live/2008-August/017204.html}}
I = {\`I,´I,\~I,\h I,\d I},
i = {\`i,´i,\~i,\h i,\d i},
O = {\`O,´O,\~O,\h O,\d O,\horn O},
o = {\`o,´o,\~o,\h o,\d o,\horn o},
U = {\`U,´U,\~U,\h U,\d U,\horn U},
u = {\`u,´u,\~u,\h u,\d u,\horn u},
Y = {\`Y,´Y,\~Y,\h Y,\d Y},
y = {\`y,´y,\~y,\h y,\d y}
\{
\}

15.5.7 EU1, EU2, TU

The EU1 (XyloTeX), EU2 (LuaTeX), and, since fontsque version 2.5, TU encodings are not well-defined in the sense that they don’t contain a fixed number of glyphs, all of which must be present. OpenType fonts may contain thousands of glyphs, but we only define those that should be present in every font (basically T1). This inheritance list should be overridden by font-specific ones.

\{ encoding = {EU1,EU2,TU} 
A = {\`A,´A,\^A,\~A,"A,\r A,\k A,\u A},
a = {\`a,´a,\^a,\~a,"a,\r a,\k a,\u a},
C = {\c C,\v C},
c = {\c c,\v c},
D = {\v D,\DH},
d = {\v d,\dj},
E = {\`E,´E,\^E,"E,\k E,\v E},
e = {\`e,´e,\^e,"e,\k e,\v e},
f = {/f _f},
g = {\u G},
I = {\`I,´I,\^I,"I,\.I},
i = {\`i,´i,\^i,"i,\.i},
j = {\j},
L = {\l L,\v L},
I = {\`I,´I,\^I,"I,\.I},
N = {\`N,\v N},
n = {\`n,\v n},
O = {\`O,´O,\~O,"O,\H O},
o = {\`o,´o,\~o,"o,\H o},
R = {\v R},
r = {\v r,\v r},
S = {\`S,\v S,\l S,\l S,\SS},
s = {\`s,\v s,\l s,\l s},
T = {\c T,\v T},
t = {\c t,\v t},
U = {\`U,´U,\~U,\H U,\r U},
u = {\`u,´u,\~u,\H u,\r u},
Y = {\`Y,Y},
y = {\`Y,Y},
Z = {\`Z,\v Z},
z = {\`z,\v z,\v z}
\}
15.5.8 Euro symbols

Make Euro symbols settings simpler.

\begin{verbatim}
\DeclareCharacterInheritance
{ encoding = U,
  family = {zpeu,zpeus,eurosans} }
\end{verbatim}

Since 2006/05/11 (that is, one week after I've added these settings, after the package had been dormant for six years!), marvosym's encoding is (correctly) U instead of OT1.

\begin{verbatim}
\DeclareCharacterInheritance
{ encoding = {OT1,U},
  family = mvs }
\end{verbatim}

15.6 Tracking

By default, we only disable the ‘f*’ ligatures, for those fonts that have any. Thus, ligatures and especially kerning for all other characters will be retained.

\begin{verbatim}
\SetTracking
[ name = default,
  no ligatures = {f} ]
\end{verbatim}

15.7 Font expansion

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

\begin{verbatim}
\SetExpansion
[ name = default ]
\end{verbatim}
\OE = 500, \oe = 700,
P = 700, p = 700,
Q = 500, q = 700,
R = 700,
S = 700, s = 700,
U = 700, u = 700,
W = 700, w = 700,
Z = 700, z = 700,
2 = 700,
3 = 700,
6 = 700,
8 = 700,
9 = 700,
}

Settings for Cyrillic T2A encoding.\footnote{Contributed by Karl Karlsson.}
\SetExpansion
{ name = T2A }
{ encoding = T2A }
{
A = 500, a = 700,
B = 700, b = 700,
C = 700, c = 700,
D = 500, d = 700,
E = 700, e = 700,
F = 700,
G = 500, g = 700,
H = 700, h = 700,
K = 700, k = 700,
M = 700, m = 700,
N = 700, n = 700,
O = 500, o = 700,
P = 700, p = 700,
Q = 500, q = 700,
R = 700,
S = 700, s = 700,
U = 700, u = 700,
W = 700, w = 700,
Z = 700, z = 700,
2 = 700,
3 = 700,
6 = 700,
8 = 700,
9 = 700,
\CYRA = 500, \cyra = 700,
\CYRB = 700, \cyrb = 700,
\CYRV = 700, \cyrv = 700,
\CYRG = 700, \cyrg = 700,
\CYRD = 700, \cyrd = 700,
\CYRE = 700, \cyre = 700,
\CYRZH = 700, \cyrzh = 700,
\CYRZ = 700, \cyrz = 700,
\CYRI = 700, \cyri = 700,
\CYRISHRT = 700, \cyrishrt = 700,
\CYRK = 700, \cyrk = 700,
\CYRL = 700, \cyr1 = 700,
\CYRM = 700, \cyrm = 700,
\CYRN = 700, \cyrn = 700,
\CYRO = 500, \cyro = 700,
\CYRP = 700, \cyrp = 700,
\CYRR = 700, \cyr = 700,
\CYRS = 700, \cyr = 700,
\CYRT = 700, \cyrt = 700,
\CYRU = 700, \cyru = 700,
\CYRF = 700, \cyrf = 700,
\CYRH = 700, \cyrh = 700,
\CYRC = 700, \cyrc = 700,
\CYRCH = 700, \cyrch = 700,
\CYRSFTN = 700, \cyrsftn = 700,
\CYREREV = 700, \cyrerev = 700,
\CYRYU = 700, \cyryu = 700,
\CYRYA = 700, \cyrya = 700

\CYRHRDSN = 700, \cyrhrdsn = 700,
\CYRERY = 700, \cyrery = 700,
\CYRSFTSN = 700, \cyrsftsn = 700,
\CYRHRDSN = 700, \cyrhrdsn = 700,
\CYRERY = 700, \cyrery = 700,
\CYRSFTSN = 700, \cyrsftsn = 700,
\CYREREV = 700, \cyrerev = 700,
\CYRYU = 700, \cyryu = 700,
\CYRYA = 700, \cyrya = 700

T5 encoding does not contain \AE, \ae, \OE and \oe.

\SetExpansion [ name = T5 ]
{ encoding = T5 }

\SetProtrusion
[ name = thanh ]
{ encoding = OT1 }

15.8 Character protrusion

For future historians, Hàn Thế Thành’s original settings (from protcode.tex, converted to microtype notation).
15.8.1 Normal

The default settings always use the most moderate value.

```
\SetProtrusion [ name = default ]
```

We also create configuration files for the fonts

* Bitstream Charter (NFSS code bch)

```
\SetProtrusion [ bch ] [ name = bch-default ]
```

* Bitstream Letter Gothic (blg)

```
\SetProtrusion [ blg ] [ name = blg-default ]
```

* Computer Modern Roman (cmr)

```
\SetProtrusion [ cmr ] [ name = cmr-default ]
```

* Adobe Garamond (pad, padx, padj)

```
\SetProtrusion [ pad ] [ name = pad-default ]
```

* Minion\footnote{Contributed by Harald Harders and Karl Karlsson.} (pmnx, pmnj)

```
\SetProtrusion [ pmn ] [ name = pmnj-default ]
```

* Palatino (ppl, pplx, pplj)

```
\SetProtrusion [ ppl ] [ name = pplj-default ]
```

* Times (ptm, ptmx, ptmj)

```
\SetProtrusion [ ptm ] [ name = ptm-default ]
```

* URW Garamond (ugm)

```
\SetProtrusion [ ugm ]
```
CONFIGURATION FILES: Character protrusion

```
5282 [ugm] { name = ugm-default }
5283 [m-t|cmr|pmn] { }
5284 [bch|blg|pad|ugm] { encoding = OT1,}
5285 [ppl|ptm] { encoding = {OT1,OT4},}
5286 [bch] family = bch
5287 [blg] family = blg
5288 [pad] family = {pad,pxd,padj}
5289 [ppl] family = {ppl,pxl,pplj}
5290 [ptm] family = {ptm,ptmx,ptmj}
5291 [ugm] family = ugm

5292 { [m-t|bch|blg|cmr|pad|pmn|ppl|ptm] A = {50,50},
5293 [ugm] A = {50,100},
5294 [pad|ptm] \AE = {50, },
5295 [ugm] \AE = {150,50},
5296 [bch|blg] B = { ,50},
5297 [ugm] C = {50, },
5298 [bch|pad|pmn] D = { ,70},
5299 [bch|pad|pmn|ugm] E = { ,70},
5300 [ugm] F = { ,50},
5301 [ugm] G = {50,50},
5302 [ugm] H = { ,150},
5303 [ugm] I = {150,150},
5304 [ugm] J = {50,50},
5305 [ugm] K = { ,50},
5306 [ugm] L = { ,100},
5307 [ugm] M = { ,70},
5308 [ugm] N = { ,150},
5309 [ugm] O = {50,50},
5310 [ugm] P = { ,120},
5311 [ugm] Q = {50,50},
5312 [ugm] R = { ,50},
5313 [ugm] S = { ,70},
5314 [ugm] T = {100,100},
5315 [ugm] U = {50,50},
5316 [ugm] V = {100,100},
5317 [ugm] W = {70,70},
5318 [ugm] X = {50,50},
5319 [ugm] Y = {70,70},
5320 [ugm] Z = {50,50},
5321 [ugm] k = { ,70},
5322 [ugm] l = { ,50},
5323 [ugm] m = { ,70},
5324 [ugm] n = { ,50},
5325 [ugm] o = { ,70},
5326 [ugm] p = { ,70},
5327 [ugm] q = { ,50},
5328 [ugm] r = { ,50},
5329 [ugm] s = { ,50},
5330 [ugm] t = { ,50},
5331 [ugm] u = { ,50},
5332 [ugm] v = { ,50},
5333 [ugm] w = { ,50},
5334 [ugm] x = { ,50},
5335 [ugm] y = { ,50},
5336 [ugm] z = { ,50},
5337 [ugm] k = { ,50},
5338 [ugm] l = { ,50},
5339 [ugm] m = { ,50},
5340 [ugm] n = { ,50},
5341 [ugm] o = { ,50},
5342 [ugm] p = { ,50},
5343 [ugm] q = { ,50},
5344 [ugm] r = { ,50},
5345 [ugm] s = { ,50},
5346 [ugm] t = { ,50},
```
CONFIGURATION FILES: Character protrusion
Greek uppercase letters are in OT1 encoding only.
\SetProtrusion
\m-t [ name = OT1-default, ]
\cmr [ name = cmr-OT1, ]
\pmn [ name = pmnj-OT1, ]
\m-t [ load = default ]
\cmr [ load = cmr-default ]
\pmn [ load = pmnj-default ]
\m-t [ encoding = OT1 ]
\cmr [ encoding = {OT1,OT4}, ]
\pmn [ encoding = OT1, ]
\cmr [ family = cmr ]
\pmn [ family = pmnj ]
\m-t [ \AE = {50, }, ]
\pmn [ \OE = {50, } ]
\cmr [ "00 = {150}, % \Gamma ]
\pmn [ "01 = {100,100}, % \Delta ]
\cmr [ "02 = {50,50}, % \Theta ]
\pmn [ "03 = {100,100}, % \Lambda ]
\cmr [ "06 = {50,50}, % \Sigma ]
\pmn [ "07 = {100,100}, % \Upsilon ]
\cmr [ "08 = {50,50}, % \Phi ]
\pmn [ "09 = {50,50} % \Psi ]

Remaining slots can be found in the source file.
\cmr 
\m-t|cmr|pmn

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For \TeX (EU1) and Lua\TeX (EU2) we simply use the T1 list as default (for now).
\cmr 
\m-t|cmr|pmn|ppl

\SetProtrusion
\m-t [ name = T1-default, ]
\bch [ name = bch-T1, ]
\blg [ name = blg-T1, ]
\cmr [ name = cmr-T1, ]
\pad [ name = pad-T1, ]
\pmn [ name = pmnj-T1, ]
\ppl [ name = ppl-T1, ]
\ptm [ name = ptm-T1, ]
\ugm [ name = ugm-T1, ]
\m-t [ load = default ]
\bch [ load = bch-default ]
\blg [ load = blg-default ]
\cmr [ load = cmr-default ]
\pad [ load = pad-default ]
\pmn [ load = pmnj-default ]
\ppl [ load = ppl-default ]
\ptm [ load = ptm-default ]
\ugm [ load = ugm-default ]
\m-t [ encoding = {T1,LY1,EU1,EU2,TU} ]
\bch [ encoding = {T1,LY1}, ]
\pad [ encoding = {T1}, ]
\bch [ family = bch ]
\blg [ family = blg ]
\cmr [ family = cmr ]
\pad [ family = {pad,padx,padj} ]
\pmn [ family = pmnj ]
\ppl [ family = {ppl,pplx,pplj} ]
\ptm [ family = {ptm,ptmx,ptmj} ]
\ugm [ family = ugm ]
\m-t
The EC fonts do something weird: they insert an implicit kern between quote and boundary character. Therefore, we must override the settings from OT1.

The \textbackslash fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

The \textbackslash modern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.
Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).\textsuperscript{20}

\begin{verbatim}
{encoding = T2A,
\family = cmr}
\{ encoding = T2A,
\family = pmn}
\SetProtrusion \[ name = T2A-default,
\family = cmr\[ name = cmr-T2A,
\family = pmn\[ name = pmnj-T2A,
\load = default \]
\load = cmr-default \]
\load = pmnj-default \}
\{ \CYRA = {50,50},
\CYRG = {,50},
\CYRK = {,50},
\CYRT = {50,50},
\CYRH = {50,50},
\CYRU = {50,50},
\CYRS = {50, },
\CYRO = {50,50},
\cyrk = { ,50},
\cyrg = { ,50},
\cyrh = {50,50},
\cyru = {50,50},
\textbackslash = {100,200}, \quotedblbase = {400,400},
\textbackslash = {200,300}, \textquotedbl = {300,300}, \textquotedblleft = {200,600},
\guillemotleft = {200,200}, \guillemotright = {200,200},
\guillemotleft = {300,200}, \guillemotright = {100,400},
\textbraceleft = {400,200}, \textbraceright = {200,400},
\textbraceleft = {200, }, \textbraceright = { ,300},
\textless = {200,100}, \textgreater = {100,200},
\textless = {100, }, \textgreater = { ,100}\}
\end{verbatim}

Settings for the QX encoding (generic and Times).\textsuperscript{21} It also includes some glyphs otherwise in TS1.

\begin{verbatim}
{encoding = QX,
\family = pmn}
\SetProtrusion \[ name = QX-default,
\family = ptm\[ name = ptm-QX,
\family = pmn\[ name = pmn-QX,
\load = default \]
\load = ptm-default \]
\{ \AE = {50, },
\* = {200,200},
\* = {100,100},
\textunderscore = {100,100},
\textbackslash = {100,200},
\quotedblbase = {400,400}\}
\end{verbatim}

\textsuperscript{20} Contributed by Karl Karlsson.
\textsuperscript{21} Contributed by Maciej Eder.
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.

Minion with lining numbers.
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

Remaining slots in the source file.

15.8.2 Italics

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

The italic glyphs of Computer Modern Roman feature a lot of side bearing, therefore almost all of them have to protrude.\textsuperscript{22}

\begin{verbatim}
\SetProtrusion
\(m-t\) [ name = OT1-it ]
\(bch\) [ name = bch-it ]
\(blg\) [ name = blg-it, load = blg-default ]
\(cmr\) [ name = cmr-it ]
\(pad\) [ name = pad-it ]
\(pmn\) [ name = pmn-it ]
\(ppl\) [ name = ppl-it ]
\(ptm\) [ name = ptm-it ]
\(ugm\) [ name = ugm-it ]
\(m-t\|bch|blg|pad|ugm\) { encoding = OT1,
\(bch|ppl|ptm\) { encoding = {OT1,OT4},
\(bch\) family = bch,
\(blg\) family = blg,
\(pad\) family = {pad,padx,padj},
\(ppl\) family = {ppl,pplx,pplj},
\(ptm\) family = {ptm,ptmx,ptmj},
\(ugm\) family = ugm,
\(m-t|bch|pad|ppl|ptm\) shape = {it,sl} }
\(blg|ugm\) shape = it }
\(cmr|pmn\) { }
\(cmr\) A = {100,100},
\(ptm\) A = {100,50},
\(pad|pmn\) A = {50, },
\(ugm\) A = { ,150},
\(ppl\) A = {50,50},
\(ptm\) \(\text{\AE}\) = {100, },
\(pad\) \(\text{\AE}\) = {50, },
\(cmr\) B = {83,-40},
\(pad|ppl|ptm\) B = {50, },
\(pmn\) B = {20,-50},
\(bch|ppl|ptm|ugm\) C = {50, },
\(cmr\) C = {165,-75},
\(pad\) C = {100, },
\(pmn\) C = {50,-50},
\(ppl\) C = {75, -28},
\(ptm\) D = {50,50},
\(pmn\) D = {20, },
\(cmr\) E = {80,-55},
\(pad|ppl|ptm\) E = {50, },
\(pmn\) E = {20,-50},
\(cmr\) F = {85,-80},
\(pad|ptm\) F = {100, },
\(pmn\) F = {10, },
\(ppl\) F = {50, },
\(bch|ppl|ptm|ugm\) G = {50, },
\(cmr\) G = {153,-15},
\(pad\) G = {100, },
\(pmn\) G = {50,-50},
\(cmr\) H = {73,-60},
\(pad|ppl|ptm\) H = {50, },
\(cmr\) I = {140,-120},
\(pad|ptm\) I = {50, },
\(pmn\) I = {20,-50},
\(cmr\) J = {135,-80},
\(pad\) J = {50, },
\(pmn\) J = {20, }.
\end{verbatim}

\textsuperscript{22} Settings contributed by Hendrik Vogt.
CONFIGURATION FILES: Character protrusion

\[
\begin{align*}
\text{J} &= \{100, \} \\
\text{K} &= \{70, -30, \} \\
\text{L} &= \{20, 50, \} \\
\text{M} &= \{67, -45, \} \\
\text{N} &= \{75, -55, \} \\
\text{O} &= \{50, \} \\
\text{P} &= \{82, -50, \} \\
\text{Q} &= \{50, \} \\
\text{R} &= \{75, 15, \} \\
\text{S} &= \{50, \} \\
\text{T} &= \{220, -85, \} \\
\text{U} &= \{50, \} \\
\text{V} &= \{100, 50, \} \\
\text{W} &= \{100, -50, \} \\
\text{X} &= \{70, -30, \} \\
\text{Y} &= \{250, -60, \} \\
\text{Z} &= \{90, -60, \} \\
\text{a} &= \{150, -10, \} \\
\text{b} &= \{170, \} \\
\text{c} &= \{173, -10, \} \\
\text{d} &= \{150, -55, \} \\
\text{e} &= \{180, \} \\
\text{f} &= \{100, -100, \}
\end{align*}
\]
CONFIGURATION FILES : Character protrusion

\{cmr\} q = \{150, -10\},
\{cmr\} h = \{100, \},
\{cmr\} i = \{210, \},
\{cmr\} j = \{-30, \},
\{cmr\} j = \{-40, \},
\{cmr\} k = \{110, -50\},
\{cmr\} l = \{240, -110\},
\{cmr\} l = \{-100, \},
\{cmr\} m = \{80, \},
\{cmr\} m = \{115, \},
\{bch\} o = \{50,50\},
\{cmr\} o = \{155, \},
\{pmn\} o = \{50, \},
\{pmn\} p = \{-50, \},
\{pmn\} q = \{50, \},
\{pmn\} q = \{170, -40\},
\{cmr\} r = \{155, -40\},
\{pmn\} r = \{50, \},
\{pmn\} s = \{130, \},
\{bch\} t = \{, 50\},
\{cmr\} t = \{230, -10\},
\{cmr\} u = \{120, \},
\{pmn\} u = \{140, -25\},
\{pmn\} u = \{50, \},
\{bch\} w = \{, 50\},
\{cmr\} w = \{98, -20\},
\{pmn\} u = \{50, \},
\{cmr\} x = \{65, -40\},
\{bch\} y = \{, 50\},
\{pmn\} y = \{130, -20\},
\{cmr\} z = \{110, -80\},
\{cmr\} 0 = \{170, -85\},
\{bch\} ptm 1 = \{150,100\},
\{cmr\} 1 = \{230,110\},
\{ptm\} 1 = \{150, \},
\{cmr\} 1 = \{50, \},
\{pmn\} 1 = \{100, \},
\{pmn\} ug m 1 = \{150,150\},
\{cmr\} 2 = \{130, -70\},
\{pad\} ptm \{50, \},
\{bch\} ptm 2 = \{-50, \},
\{bch\} ptm 3 = \{50, \},
\{cmr\} ptm 3 = \{140, -70\},
\{pmn\} ptm 3 = \{-100, \},
\{pmn\} ptm 3 = \{100,50\},
\{bch\} ptm 4 = \{100, \},
\{cmr\} ptm 4 = \{130,80\},
\{pad\} ptm 4 = \{150, \},
\{pmn\} ptm 4 = \{50, \},
\{pmn\} ptm 5 = \{160, \},
\{ptm\} 5 = \{50, \},
\{bch\} 6 = \{50, \}
\{cmr\} 6 = \{175, -30\},
\{bch\} pad ptm 7 = \{100, \},
\{cmr\} 7 = \{250, -150\},
\{pmn\} 8 = \{20, \},
\{pmn\} 8 = \{50, \},
\{pmn\} 8 = \{130, -40\},
\{cmr\} 9 = \{155, -80\},
\{pmn\} 9 = \{50, \},
\{cmr\} 9 = \{155, -80\},
\{pmn\} 9 = \{, 500\},
\{pmn\} 9 = \{400,600\},
\{pmn\} 9 = \{, 700\},
\{pmn\} 9 = \{300,500\},
\{pmn\} 9 = \{, 500\}.
CONFIGURATION FILES : Character protrusion

5944 \emph{cmr} \{,\} = \{( .450),
5945 \emph{bch} \emph{ugm} \{,\} = \{( .600),
5946 \emph{ptm} \{,\} = \{( .700),
5947 \emph{m-t} \emph{cmr} \emph{pad} \emph{ppl} \{,\} = \{( .300),
5948 \emph{bch} \emph{ugm} \{,\} = \{( .400),
5949 \emph{pmm} \{,\} = \{( .200),
5950 \emph{ptm} \{,\} = \{( .500),
5951 \emph{m-t} \emph{cmr} \emph{pad} \emph{ppl} \{,\} = \{( .300),
5952 \emph{bch} \emph{ugm} \{,\} = \{( .400),
5953 \emph{pmm} \{,\} = \{( .400),
5954 \emph{ptm} \{,\} = \{( .500),
5955 \emph{ptm} \{,\} = \{( .100),
5956 \emph{bch} \{,\} = \{( .200),
5957 \emph{ptm} \{,\} = \{( .100),
5958 \emph{ppl} \{,\} = \{( .300),
5959 \emph{pmm} \{,\} = \{( .400),
5960 \emph{m-t} \emph{pad} \emph{ppl} \emph{ptm} \& = \{( 50,50),
5961 \emph{bch} \& = \{( .80),
5962 \emph{cmr} \& = \{( 130,30),
5963 \emph{ugm} \& = \{( 50,100),
5964 \emph{m-t} \emph{pad} \emph{pmm} \\% = \{( 100,100),
5965 \emph{cmr} \\% = \{( 180,50),
5966 \emph{bch} \\% = \{( 50,50),
5967 \emph{ppl} \emph{ptm} \\% = \{( 100,100),
5968 \emph{ugm} \\% = \{( 100,50),
5969 \emph{m-t} \emph{pmm} \emph{ppl} \* = \{( 200,200),
5970 \emph{bch} \* = \{( 300,200),
5971 \emph{cmr} \* = \{( 380,20),
5972 \emph{pad} \* = \{( 500,100),
5973 \emph{ptm} \emph{ugm} \* = \{( 400,200),
5974 \emph{m-t} \emph{pmm} \emph{ppl} \+ = \{( 150,200),
5975 \emph{cmr} \+ = \{( 180,200),
5976 \emph{bch} \emph{ugm} \+ = \{( 250,250),
5977 \emph{pad} \emph{ptm} \+ = \{( 250,200),
5978 \emph{m-t} \emph{pad} \emph{pmm} \emph{ppl} \0 = \{( 50,50),
5979 \emph{bch} \0 = \{( 80,50),
5980 \emph{cmr} \0 = \{( 180,10),
5981 \emph{pmm} \0 = \{( 150,150),
5982 \emph{m-t} \emph{bch} \emph{ugm} \textendash = \{( 150,150),
5983 \emph{cmr} \emph{pad} \emph{pmm} \emph{ppl} \emph{ptm} \textendash = \{( 200,150),
5984 \emph{ugm} \textemdash = \{( 200,200),
5985 \emph{m-t} \emph{bch} \emph{pad} \emph{pmm} \emph{ppl} \emph{ptm} \emph{ugm} \textendash = \{( 200,200),
5986 \emph{cmr} \textendash = \{( 100,100),
5987 \emph{bch} \textendash = \{( .150),
5988 \emph{pmm} \textendash = \{( 100,150),
5989 \emph{m-t} \textendash = \{( 300,300),
5990 \emph{bch} \textemdash = \{( 300,400),
5991 \emph{pmm} \textemdash = \{( 200,300),
5992 \emph{ppl} \textemdash = \{( 500,300),
5993 \emph{ptm} \textemdash = \{( 500,500),
5994 \emph{ugm} \textemdash = \{( 400,700),
5995 \emph{big} \textemdash = \{( 6,300),
5996 \emph{m-t} \emph{pmm} \\textendash = \{( 200,200), \textendash = \{( 150,150),
6000 \emph{bch} \textendash = \{( 200,300), \textendash = \{( 150,200),
6001 \emph{cmr} \textendash = \{( 500,300), \textendash = \{( 400,170),
6002 \emph{pad} \emph{ppl} \emph{ptm} \emph{ugm} \textendash = \{( 300,300), \textendash = \{( 200,200),
6003 \emph{m-t} \emph{bch} \emph{pad} \emph{pmm} \textemdash = \{( 400,200), \textemdash = \{( 400,200),
6004 \emph{big} \textemdash = \{( 400,400), \textemdash = \{( 400,400),
6005 \emph{cmr} \textemdash = \{( 800,200), \textemdash = \{( 800,200),
6006 \emph{pad} \textemdash = \{( 800,200), \textemdash = \{( 800,200),
6007 \emph{ppl} \textemdash = \{( 700,400), \textemdash = \{( 700,400),
6008 \emph{ptm} \textemdash = \{( 800,500), \textemdash = \{( 800,500),
\textbf{CONFIGURATION FILES}:

Character protrusion

\begin{verbatim}
6009 \[m|bch|pmn]\textbackslash textquotedblleft = \{400,200\}, \textbackslash textquotedblright = \{400,200\}
6010 \[big]\textbackslash textquotedblleft = \{300,300\}
6011 \[cmr]\textbackslash textquotedblleft = \{540,100\}, \textbackslash textquotedblright = \{500,100\}
6012 \[pad]\textbackslash textquotedblleft = \{700,200\}, \textbackslash textquotedblright = \{700,200\}
6013 \[ppl]\textbackslash textquotedblleft = \{500,300\}, \textbackslash textquotedblright = \{500,300\}
6014 \[ptm]\textbackslash textquotedblleft = \{700,400\}, \textbackslash textquotedblright = \{700,400\}
6015 \[ugm]\textbackslash textquotedblleft = \{600,200\}, \textbackslash textquotedblright = \{600,200\}
6016 }
6017 6018 \[\ast cmr|pmn]\textbackslash SetProtrusion
6019 \[cmr]\[ name = cmr-it-OT1,
6020 \[pmn]\[ name = pmnj-it-OT1,
6021 \[cmr]\[ load = cmr-it \]
6022 \[pmn]\[ load = pmnj-it \]
6023 \[cmr]\[ family = cmr,
6024 \[pmn]\[ family = pmnj,
6025 \[cmr]\[ shape = it \]
6026 \[pmn]\[ shape = \{it,sl\} \]
6027 \}
6028 \}
6029 \[cmr]\ \AE = \{100, \}
6030 \[pmn]\ \AE = \{-50, \}
6031 \[cmr]\ \OE = \{100, \}
6032 \[pmn]\ \OE = \{50, \}
6033 \[cmr]\ *00 = \{200,150\}, \% \Gamma
6034 \[pmn]\ *01 = \{150,100\}, \% \Delta
6035 \[cmr]\ *02 = \{150, 50\}, \% \Theta
6036 \[pmn]\ *03 = \{150, 50\}, \% \Lambda
6037 \[cmr]\ *04 = \{100,100\}, \% \Xi
6038 \[pmn]\ *05 = \{100,100\}, \% \Pi
6039 \[cmr]\ *06 = \{100, 50\}, \% \Sigma
6040 \[pmn]\ *07 = \{200,150\}, \% \Upsilon
6041 \[cmr]\ *08 = \{150,100\}, \% \Phi
6042 \[pmn]\ *09 = \{150,100\}, \% \Psi
6043 \[cmr]\ *0A = \{ 50, 50\} % \Omega
6044 \}
6045 \}
6046 \[cmr]\ \[\textbackslash SetProtrusion
6047 \[m-t|bch|cmr|pad|pmn|ppl]\[ encoding = \{T1,LY1\},
6048 \[big|ptm|ugm]\[ encoding = T1,
6049 \[bch]\ family = bch,
6050 \[big]\ family = big.
\end{verbatim}
6074 \{ cmr \} \textfamily = \textfamily{cmr},
6075 \{ pmn \} \textfamily = \textfamily{pmn},
6076 \{ pad \} \textfamily = \textfamily{pad}, \textfamily{padx}, \textfamily{padj},
6077 \{ ppl \} \textfamily = \textfamily{ppl}, \textfamily{pplx}, \textfamily{pplj},
6078 \{ ptm \} \textfamily = \textfamily{ptm}, \textfamily{ptmx}, \textfamily{ptmj},
6079 \{ ugm \} \textfamily = \textfamily{ugm},
6080 \{ m-t \} \textfamily{bch} \{ pmn \} \textfamily{ppl} \{ ptm \} \textshape = \{ \textit{s}, \textit{sl} \}
6081 \{ big \} \textfamily{cmr} \textfamily{ugm} \textshape = \textit{it}
6082 \{ }
6083 \{ m-t \} \textfamily{bch} \{ pmn \} \textshape = \{ .100 \},
6084 \{ big \} \textshape = \{ 0,300 \},
6085 \{ cmr \} \textfamily{ugm} \textshape = \{ 100,200 \},
6086 \{ pad \} \textfamily{ppl} \{ ptm \} \textshape = \{ 100,100 \},
6087 \{ big \} \textshape = \{ 400,600 \},
6088 \{ big \} \textshape = \{ 300,500 \},
6089 \{ cmr \} \textshape{\AE} = \{ 100 \},
6090 \{ pmn \} \textshape{\AE} = \{ -50 \},
6091 \{ bch \} \textfamily{pmn} \textshape{\OE} = \{ 50 \},
6092 \{ cmr \} \textshape{\OE} = \{ 100 \},
6093 \{ pmn \} \textshape{\OE} = \{ -.100 \}, \% \textshape{ffl}
6094 \{ cmr \} \textfamily{ptm} \textshape{156} = \{ 100, \}, \% \textshape{IJ}
6095 \{ pad \} \textshape{156} = \{ 50, \}, \% \textshape{Ij}
6096 \{ pmn \} \textshape{156} = \{ 20, \}, \% \textshape{IJ}
6097 \{ pmn \} \textshape{188} = \{ -.30, \% \textshape{iJ}
6098 \{ pmn \} \textshape{\t} = \{ .100 \},
6099 \{ m-t \} \textfamily{pad} \textfamily{ppl} \textfamily{ptm} \textshape{\textbackslash slash} = \{ 100,200 \},
6100 \{ cmr \} \textfamily{ugm} \textshape{\textbackslash slash} = \{ 300,300 \},
6101 \{ bch \} \textfamily{\textbackslash slash} = \{ 150,150 \},
6102 \{ pmn \} \textshape{\textbackslash slash} = \{ 100,150 \},
6103 \{ ugm \} \textshape{\textbar} = \{ 200,200 \},
6104 \{ cmr \} \textshape{\textvisiblespace} = \{ 500,300 \},
6105 \{ big \} \textshape{\textquotedbl} = \{ 400,400 \}, \textshape{\textquotedblright} = \{ 400,400 \},
6106 \{ big \} \textshape{\textquotedbl{} = \{ 300,300 \}, \textshape{\textquotedblright} = \{ 300,300 \},
6107 \{ big \} \textshape{\textquotedbl} = \{ 300,300 \}, \textshape{\textquotedblright} = \{ 300,300 \},
6108 \{ m-t \} \textfamily{ptm} \textshape{\textquotesingle base} = \{ 300,700 \}, \textshape{\textquotedblbase} = \{ 400,500 \},
6109 \{ cmr \} \textshape{\textquotesingle base} = \{ 300,700 \}, \textshape{\textquotedblbase} = \{ 200,600 \},
6110 \{ bch \} \textfamily{pmn} \textshape{\textquotesingle base} = \{ 200,500 \}, \textshape{\textquotedblbase} = \{ 150,500 \},
6111 \{ pad \} \textfamily{ppl} \textshape{\textquotesingle base} = \{ 500,500 \}, \textshape{\textquotedblbase} = \{ 400,400 \},
6112 \{ ugm \} \textshape{\textquotesingle base} = \{ 300,700 \}, \textshape{\textquotedblbase} = \{ 300,500 \},
6113 \{ m-t \} \textfamily{ppl} \textfamily{ptm} \textshape{\textguillemotleft} = \{ 400,400 \}, \textshape{\textguillemotright} = \{ 300,500 \},
6114 \{ bch \} \textfamily{pmn} \textshape{\textguillemotleft} = \{ 300,400 \}, \textshape{\textguillemotright} = \{ 200,500 \},
6115 \{ cmr \} \textshape{\textguillemotleft} = \{ 500,300 \}, \textshape{\textguillemotright} = \{ 400,400 \},
6116 \{ pad \} \textshape{\textguillemotleft} = \{ 400,400 \}, \textshape{\textguillemotright} = \{ 300,500 \},
6117 \{ ugm \} \textshape{\textguillemotleft} = \{ 400,400 \}, \textshape{\textguillemotright} = \{ 300,600 \},
6118 \{ m-t \} \textfamily{ppl} \textshape{\textguillemotleft} = \{ 300,300 \}, \textshape{\textguillemotright} = \{ 300,300 \},
6119 \{ bch \} \textfamily{pmn} \textshape{\textguillemotleft} = \{ 200,300 \}, \textshape{\textguillemotright} = \{ 150,400 \},
6120 \{ cmr \} \textshape{\textguillemotleft} = \{ 400,100 \}, \textshape{\textguillemotright} = \{ 200,300 \},
6121 \{ pad \} \textshape{\textguillemotleft} = \{ 300,300 \}, \textshape{\textguillemotright} = \{ 200,400 \},
6122 \{ ptm \} \textshape{\textguillemotleft} = \{ 300,400 \}, \textshape{\textguillemotright} = \{ 200,400 \},
6123 \{ ugm \} \textshape{\textguillemotleft} = \{ 300,400 \}, \textshape{\textguillemotright} = \{ 300,400 \},
6124 \{ m-t \} \textfamily{ppl} \textfamily{ugm} \textshape{\textexclamdown} = \{ 100, \}, \textshape{\textquestiondown} = \{ 200, \},
6125 \{ cmr \} \textfamily{ptm} \textshape{\textexclamdown} = \{ 200, \}, \textshape{\textquestiondown} = \{ 200, \},
6126 \{ pmn \} \textshape{\textexclamdown} = \{ -50, \}, \textshape{\textquestiondown} = \{ -50, \},
6127 \{ m-t \} \textfamily{ppl} \textfamily{ugm} \textshape{\textbraceleft} = \{ 200,100 \}, \textshape{\textbraceright} = \{ 200,200 \},
6128 \{ bch \} \textfamily{pmn} \textshape{\textbraceleft} = \{ 200, \}, \textshape{\textbraceright} = \{ .200 \},
6129 \{ cmr \} \textfamily{pad} \textfamily{ptm} \textshape{\textbraceright} = \{ 400,100 \}, \textshape{\textbraceright} = \{ 200,200 \},
6130 \{ bch \} \textfamily{pmn} \textshape{\textless} = \{ 100, \}, \textshape{\textgreater} = \{ .100, \},
6131 \{ cmr \} \textfamily{pad} \textfamily{ptm} \textshape{\textless} = \{ 300,100 \}, \textshape{\textgreater} = \{ 200,100 \},
6132 \{ pmn \} \textshape{\textvisiblespace} = \{ 100,100 \},
6133 \{ }
6134
6135 \{ m-t \} \textfamily{cmr} \textfamily{pmn}
6136 \SetProtrusion
6137 \{ m-t \} \textfamily{cmr} \textfamily{pmn}
6138 \{ m-t \} \textfamily{cmr} \textfamily{pmn}
CONFIGURATION FILES: Character protrusion

6139 (pmn)  [ name = pmnj-it-T2A,
6140 (m-t)   load = OT1-it ]
6141 (cmr)   load = cmr-it ]
6142 (pmn)   load = pmnj-it ]
6143 (encoding = T2A,
6144 (cmr)   family = cmr,
6145 (pmn)   family = pmnj,
6146 (m-t|pmn) shape = {it,s1} ]
6147 (cmr)   shape = it }
6148 {
6149 (cmr) \CYRA = (100,50),
6150 (pmn) \CYRA = (50, ),
6151 (cmr) \CYRB = (50, ),
6152 (cmr) \CYRV = (50, ),
6153 (pmn) \CYRV = (20,-50),
6154 (cmr) \CYRG = (100, ),
6155 (pmn) \CYRG = (10, ),
6156 (cmr) \CYRD = (50, ),
6157 (cmr) \CYRE = (50, ),
6158 (cmr) \CYRZ = (20,-50),
6159 (cmr) \CYRZ = (50, ),
6160 (cmr) \CYRI = (50, ),
6161 (pmn) \CYRI = ( ,30),
6162 (cmr) \CYRISHRT = (50, ),
6163 (cmr) \CYRSFTSN = (50, ),
6164 (cmr) \CYREREV = (50, ),
6165 (cmr) \CYRYU = (50, ),
6166 (cmr) \CYRYA = (50, ),
6167 (cmr) \CYRHRDSN = (100, ),
6168 (pmn) \cyrr = (-50, ),
6169 (m-t|pmn) _ = ( ,100),
6170 (cmr) \textbackslash = (300,300), \quotedblbase = {200,600},
Slanted is very similar to italic.
\textendash = \{400,\}, \textemdash = \{300,\}
}

\SetProtrusion
[ name = lmr-it-T1, 
load = cmr-it-T1 ]
( encoding = \{TL,LY1\},
family = lmr,
shape = \{it,sl\} )

\\textquotedblleft = \{200\}, \textquotedblright = \{200\},
\\quotesinglbase = \{400\}, \quotedblbase = \{500\}
}

Oldstyle numerals are slightly different.

\SetProtrusion
[ name = cmr(oldstyle)-it, 
load = cmr-it-T1 ]
( encoding = \{T1,\},
family = \{hfor,cmor\},
shape = \{it,sl\} )

1 = \{(250,50),
2 = \{(150,-100),
3 = \{(100,-50),
4 = \{(150,150),
6 = \{(200,\)},
7 = \{(200,50),
8 = \{(150,-50),
9 = \{(100,50)
}

\SetProtrusion
[ name = pmnx-it, 
load = pmnj-it ]
( encoding = \{OT1,\},
family = pmnx,
shape = \{it,sl\} )

1 = \{(100,150)
}

\SetProtrusion
[ name = pmnx-it-T1, 
load = pmnj-it-T1 ]
( encoding = \{T1,LY1\},
family = pmnx,
shape = \{it,sl\} )

1 = \{(100,150)
}

\SetProtrusion
[ name = pmnx-it-T2A, 
load = pmnj-it-T2A ]
( encoding = \{T2A,\},
family = pmnx,
shape = \{it,sl\} )

1 = \{(100,150)
}
15.8.3 Small caps

Small caps should inherit the values from their big brothers. Since values are relative to character width, we don’t need to adjust them any further (but we have to reset some characters).
CONFIGURATION FILES: Character protrusion 171

6452 (pmn) family = pmnj.
6453 (ppl) family = {ppl,pplx,pplj}.
6454 (ptm) family = {ptm,ptmx,ptmj}.
6455 shape = sc
6456
6457 a = (50,50).
6458 (cmr|pad|ppl|ptm) \ae = (50, ).
6459 (bch|pmn) c = (50, ),
6460 (bch|pad|pmn) d = ( .50),
6461 (m-t|bch|cmr|pad|ppl|ptm) f = ( .50),
6462 (bch|pad|pmn) g = (50, ),
6463 (m-t|cmr|pad|ppl|ptm) j = (50, ).
6464 (bch) j = (100, ),
6465 (m-t|bch|cmr|pad|ppl|ptm) l = ( .50),
6466 (ptm) 1 = ( .80),
6467 (m-t|bch|cmr|pad|ppl|ptm) 013 = ( .50), \% fl
6468 (ptm) 013 = ( .80), \% fl
6469 (bch|pad|pmn) p = (50,50),
6470 (pad|pmn) \oe = (50, ),
6471 (bch|pmn) q = (50,70),
6472 (ptm) q = (0, ).
6473 (m-t|cmr|pad|ppl|ptm) r = ( ,0),
6474 (bch) t = (50,50),
6475 (m-t|bch|cmr|pad|ppl|ptm) y = (50,50)
6476 (ptm) y = (80,80)
6477
6478 \SetProtrusion
6479
6480 \SetProtrusion
6481 (m-t) [ name = T1-sc, 
6482 (bch) [ name = bch-sc-T1, 
6483 (cmr) [ name = cmr-sc-T1, 
6484 (pad) [ name = pad-sc-T1, 
6485 (pmn) [ name = pmnj-sc-T1, 
6486 (ppl) [ name = ppl-sc-T1, 
6487 (ptm) [ name = ptm-sc-T1, 
6488 (m-t) [ load = T1-default ]
6489 (bch) load = bch-T1 ]
6490 (cmr) load = cmr-T1 ]
6491 (pad) load = pad-T1 ]
6492 (pmn) load = pmnj-T1 ]
6493 (ppl) load = ppl-T1 ]
6494 (ptm) load = ptm-T1 ]
6495 [ encoding = {T1,LY1},
6496 (bch) family = bch,
6497 (cmr) family = cmr,
6498 (pad) family = {pad,padx,padj},
6499 (pmn) family = pmnj,
6500 (ppl) family = {ppl,pplx,pplj},
6501 (ptm) family = {ptm,ptmx,ptmj},
6502 shape = sc ]
6503
6504 a = (50,50),
6505 (cmr|pad|ppl|ptm) \ae = (50, ).
6506 (bch|pmn) c = (50, ),
6507 (bch|pad|pmn) d = ( .50),
6508 (m-t|bch|cmr|pad|ppl|ptm) f = ( .50),
6509 (bch|pad|pmn) g = (50, ),
6510 (m-t|cmr|pad|ppl|ptm) j = (50, ).
6511 (bch) j = (100, ),
6512 (m-t|bch|cmr|pad|ppl) l = ( .50),
6513 (ptm) 1 = ( .80),
6514 (m-t|bch|cmr|pad|ppl|ptm) 029 = ( .50), \% fl
6515 (ptm) 029 = ( .80), \% fl
6516 (bch|pad|pmn) o = (50,50),
CONFIGURATION FILES : Character protrusion

6517 \{bch|pad|pmn\} \oe = \{50, 0\},
6518 \{ppl\} p = \{ 0, 0\},
6519 \{bch|pad|pmn\} q = \{50,70\},
6520 \{ppl\} q = \{ 0, \},
6521 \{m-t|cmr|pad|pmn|ppl|ptm\} r = \{ , 0\},
6522 \{90\} t = \{50,50\},
6523 \{m-t|bch|cmr|pad|pmn|ppl\} y = \{50,50\}
6524 \{ptm\} y = \{80,80\}
6525 \}
6526
6527 \{/!(blg|ugm)\}
6528 \{/m-t|cmr\}
6529 \SetProtrusion
6530 \{m-t\} \{ name = T2A-sc,  
6531 \{cmr\} \{ name = cmr-sc-T2A,  
6532 \{m-t\} \{ load = T2A-default \}
6533 \{cmr\} \{ load = cmr-T2A \}
6534 \{ encoding = T2A,  
6535 \{cmr\} \{ family = cmr,  
6536 \{ shape = sc \}
6537 \}
6538 \\cyra = \{50,50\},
6539 \\cyrg = \{ ,50\},
6540 \\cyrt = \{50,50\},
6541 \\cryy = \{ ,50\}
6542 \}
6543
6544 \{/m-t|cmr\}
6545 \{m-t\}
6546 \SetProtrusion
6547 \{ name = QX-sc,  
6548 \{ load = QX-default \}
6549 \{ encoding = QX,  
6550 \{ shape = sc \}
6551 \}
6552 \a = \{50,50\},
6553 \f = \{ ,50\},
6554 \j = \{50, \},
6555 \l = \{ ,50\},
6556 \013 = \{ ,50\}, \% fl
6557 \r = \{ , 0\},
6558 \t = \{50,50\},
6559 \y = \{50,50\}
6560 \}
6561
6562 \{/m-t\}
6563 \{cmr|bch\}
6564 \SetProtrusion
6565 \{bch\} \{ name = bch-sc-T5,  
6566 \{bch\} \{ load = bch-T5 \}
6567 \{cmr\} \{ name = cmr-sc-T5,  
6568 \{cmr\} \{ load = cmr-T5 \}
6569 \{ encoding = T5,  
6570 \{bch\} \{ family = bch,  
6571 \{cmr\} \{ family = cmr,  
6572 \{ shape = sc \}
6573 \}
6574 \a = \{50,50\},
6575 \c = \{50, \},
6576 \d = \{ ,50\},
6577 \f = \{ ,50\},
6578 \g = \{50, \},
6579 \j = \{100, \},
6580 \j = \{50, \},
6581 \l = \{ ,50\},
15.8.4 Italic small caps

Minion provides real small caps in italics. The slantsc package calls them scit, Philipp Lehman's fontinstallationguide suggests si.
CONFIGURATION FILES: Character protrusion

```
s = (20, -30),
t = (70, ),
u = (50, -50),
v = (100, ),
w = (100, ),
y = (50, ),
z = (, -50)
```

```chisel
\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),
j = (20, 0),
k = (20, ),
l = (20, 50),
m = (, -30),
n = (, -30),
o = (50, ),
\oe = (50, -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)
}
```

```chisel
\SetProtrusion
[ name = pmnx-scit, load = pmnj-scit ]
{ encoding = OT1, family = pmnx, shape = {scit,si} }
{
  1 = (100, 150)
}
```

```chisel
\SetProtrusion
[ name = pmnx-scit-T1, load = pmnj-scit-T1 ]
{ encoding = {T1,LY1}, family = pmnx, shape = {scit,si} }
```
Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?
6765 (bch|pad|pmm) \texttrbrackdbl = ( 100, ),
6766 (big) \texttrbrackdbl = ( 200, ),
6767 (pmn) \textasciigrave = ( 200, 500 ),
6768 (bch|big|cmr|pad|pmm) \texttillow = ( 200, 250 ),
6769 (pmn) \textasciiverte = ( 300, 400 ),
6770 (pmn) \textasciicaron = ( 300, 400 ),
6771 (pmn) \textacutes = ( 200, 300 ),
6772 (pmn) \textguessed = ( 150, 300 ),
6773 (bch|pmn|ugm) \textdagger = ( 80, 80 ),
6774 (big) \textdagger = ( 200, 200 ),
6775 (cmr|pad) \textdagger = ( 100, 100 ),
6776 (ptm) \textdagger = ( 150, 150 ),
6777 (big) \textdaggerdbl = ( 150, 150 ),
6778 (cmr|pad|pmm) \textdaggerdbl = ( 80, 80 ),
6779 (ptm) \textdaggerdbl = ( 100, 100 ),
6780 (bch) \textbar = ( 100, 100 ),
6781 (big|ugm) \textbar = ( 150, 150 ),
6782 (bch) \textbullet = ( 200, 200 ),
6783 (big) \textbullet = ( 400, 500 ),
6784 (cmr|pad|pmm) \textbullet = ( 100, ),
6785 (ptm) \textbullet = ( 150, 150 ),
6786 (ugm) \textbullet = ( 50, 100 ),
6787 (bch|cmr|pmm) \textcelsius = ( 50, ),
6788 (pad) \textcelsius = ( 80, ),
6789 (bch) \textflorin = ( 50, 50 ),
6790 (big) \textflorin = ( 100, 100 ),
6791 (pad|ugm) \textflorin = ( 100, ),
6792 (pmn) \textflorin = ( 50, 100 ),
6793 (ptm) \textflorin = ( 50, 70 ),
6794 (cmr) \textcolummonetary = ( 50, ),
6795 (pad|pmm) \textcolummonetary = ( 50, ),
6796 (pmn) \textinterrobang = ( 100, ),
6797 (pmn) \textinterrobang = ( 100, ),
6798 (m-t|pad|ptm) \texttrademark = ( 100, 100 ),
6799 (big) \texttrademark = ( 150, 150 ),
6800 (big|cmr|ppl) \texttrademark = ( 200, 200 ),
6801 (pmn) \texttrademark = ( 50, 50 ),
6802 (ugm) \texttrademark = ( 100, 150 ),
6803 (bch|ugm) \textcent = ( 50, ),
6804 (ptm) \textcent = ( 100, 100 ),
6805 (bch) \textsterling = ( 50, ),
6806 (ugm) \textsterling = ( 50, ),
6807 (bch) \textbrokenbar = ( 200, 200 ),
6808 (big) \textbrokenbar = ( 250, 250 ),
6809 (ugm) \textbrokenbar = ( 200, 300 ),
6810 (pmn) \textasciido = ( 300, 400 ),
6811 (m-t|bch|cmr|pad|ptm|ugm) \textcopyright = ( 100, 100 ),
6812 (pmn) \textcopyright = ( 100, 150 ),
6813 (ppl) \textcopyright = ( 200, 200 ),
6814 (bch|cmr|ugm) \textordfeminine = ( 100, 200 ),
6815 (pad|pmm) \textordfeminine = ( 200, 200 ),
6816 (bch|cmr|pad|pmm|ugm) \textnot = ( 200, ),
6817 (big) \textnot = ( 200, 100 ),
6818 (m-t|bch|cmr|pad|ptm|ugm) \textregistered = ( 100, 100 ),
6819 (pmn) \textregistered = ( 50, 150 ),
6820 (ppl) \textregistered = ( 200, 200 ),
6821 (pmn) \textasciimacron = ( 150, 200 ),
6822 (m-t|ppl|ptm) \textdegree = ( 300, 300 ),
6823 (bch) \textdegree = ( 150, 200 ),
6824 (big|ugm) \textdegree = ( 200, 200 ),
6825 (cmr|pad) \textdegree = ( 400, 400 ),
6826 (pmn) \textdegree = ( 150, 400 ),
6827 (bch|cmr|pad|pmm|ugm) \textpm = ( 150, 200 ),
6828 (big) \textpm = ( 100, 100 ),
6829 (ptm) \textpm = ( 50, 80 ).
CONFIGURATION FILES: Character protrusion

```
6830 (bch|blg|ugm) \texttwosuperior = \{100,200\},
6831 (cmr) \texttwosuperior = \{50,100\},
6832 (pad) \texttwosuperior = \{200,200\},
6833 (ptm) \texttwosuperior = \{50,50\},
6834 (bch|blg|ugm) \textthreesuperior = \{100,200\},
6835 (cmr) \textthreesuperior = \{50,100\},
6836 (pad) \textthreesuperior = \{200,200\},
6837 (ptm) \textthreesuperior = \{50,50\},
6838 (pmn) \textasciicircum = \{300,400\},
6839 (bch|ugm) \textmu = \{,100\},
6840 (bch|pad) \textparagraph = \{,100\},
6841 (blg) \textparagraphcentered = \{300,400\},
6842 (cmr) \textparagraphcentered = \{300,300\},
6843 (pad) \textparagraphcentered = \{200,500\},
6844 (pmn) \textparagraphcentered = \{200,500\},
6845 (bch|blg|ugm) \textasciicircum = \{100,200\},
6846 (cmr|pad) \textasciicircum = \{100,200\},
6847 (ptm) \textasciicircum = \{50,100\},
6848 (blg|cmr|pad|pmn) \textasciicircum = \{200,200\},
6849 (cmr|pad|pmn) \textasciicircum = \{200,100\},
6850 (bch|pad|pmn) \textasciicircum = \{150,200\},
6851 (pmn) \textasciicircum = \{150,200\},
6852 (ugm) \textasciicircum = \{150,200\},
6853 (bch|pad|pmn) \textasciicircum = \{150,200\},
6854 (blg) \textasciicircum = \{150,200\},
6855 (cmr) \textasciicircum = \{150,200\},
6856 (pad) \textasciicircum = \{150,200\},
6857 (pmn) \textasciicircum = \{150,200\},
6858 (ugm) \textasciicircum = \{150,200\},
6859 (bch|pad|pmn) \textasciicircum = \{150,200\},
6860 (blg) \textasciicircum = \{150,200\},
6861 (cmr) \textasciicircum = \{150,200\},
6862 (pad) \textasciicircum = \{150,200\},
6863 (pmn) \textasciicircum = \{150,200\},
6864 (ugm) \textasciicircum = \{150,200\},
6865 (bch|pad|pmn) \textasciicircum = \{150,200\},
6866 (blg) \textasciicircum = \{150,200\},
6867 (cmr) \textasciicircum = \{150,200\},
6868 (pad) \textasciicircum = \{150,200\},
6869 (pmn) \textasciicircum = \{150,200\},
6870 (ugm) \textasciicircum = \{150,200\},
6871 (bch|pad|pmn) \textasciicircum = \{150,200\},
6872 (blg) \textasciicircum = \{150,200\},
6873 (cmr) \textasciicircum = \{150,200\},
6874 (pad) \textasciicircum = \{150,200\},
6875 (pmn) \textasciicircum = \{150,200\},
6876 (ugm) \textasciicircum = \{150,200\},
6877 \{ encoding = TS1,\}
6878 (bch) \textasciicircum = \{cmr,\}
6879 (blg) \textasciicircum = \{pad,\}
6880 (cmr) \textasciicircum = \{pmn,\}
6881 (pad) \textasciicircum = \{ugm,\}
6882 \{ugm\} \textasciicircum = \{\}
6883 \{ugm\} \textasciicircum = \{\}
6884 \{ encoding = TS1,\}
6885 \{ cmr\} \textasciicircum = \{300,600\},
6886 \{ pad\} \textasciicircum = \{400,400\},
6887 \{ cmr\} \textasciicircum = \{300,600\},
6888 \{ pad\} \textasciicircum = \{300,600\},
6889 \{ pmn\} \textasciicircum = \{300,600\},
6890 \textasciicircum = \{200,200\},
6891 \{ cmr\} \textasciicircum = \{150,150\},
6892 \{ pmn\} \textasciicircum = \{200,200\},

Remaining slots in the source file.

```
\textquotesingle{}single = \{600, 300\},
\textquotesingle{}double = \{800, 100\},
\textquotesingle{}emdash = \{300, 200\},
\textquotesingle{}enquote = \{500, 500\},
\textasteriskcentered = \{300, 200\},
\textasteriskcentered = \{500, 100\},
\textasteriskcentered = \{200, 300\},
\textasteriskcentered = \{300, 150\},
\textfrac{\text solidsus}{} = \{-200, -200\},
\textsolidus = \{100, 50\},
\textsolidus = \{100, \},
\textsolidus = \{50, \},
\textsolidus = \{-50, \},
\textsolidus = \{100, 50\},
\textsolidus = \{-100, \},
\textsolidus = \{50, 50\},
\textsolidus = \{50, 80\},
\textsolidus = \{50, \},
\textsolidus = \{20, \},
\textsolidus = \{400, \},
\textsolidus = \{400\},
\textsolidus = \{300, 300\},
\textsolidus = \{200,200\},
\textsolidus = \{250, 300\},
\textsolidus = \{100, \},
\textsolidus = \{100,\},
\textsolidus = \{300, 300\},
\textsolidus = \{200,250\},
\textsolidus = \{300, 300\},
\textsolidus = \{150,300\},
\textsolidus = \{200,300\},
\textsolidus = \{150,300\},
\textsolidus = \{100,100\},
\textsolidus = \{200,100\},
\textsolidus = \{80, 80\},
\textsolidus = \{80, \},
\textsolidus = \{80, \},
\textsolidus = \{80, 50\},
\textsolidus = \{80,50\},
\textsolidus = \{150,150\},
\textsolidus = \{200,100\},
\textsolidus = \{300, \},
\textsolidus = \{50,70\},
\textsolidus = \{50, 100\},
\textsolidus = \{100, \},
\textsolidus = \{200, \},
\textsolidus = \{50, -50\},
\textsolidus = \{100, \},
\textsolidus = \{50, \},
\textsolidus = \{100, \},
\textsolidus = \{50,100\},
\textsolidus = \{150, \},
\textsolidus = \{100, \},
\textsolidus = \{50, -50\},
\textsolidus = \{200, \},
\textsolidus = \{200, \},
\textsolidus = \{200, \},
\textsolidus = \{150,50\},
\textsolidus = \{50, \},
\textsolidus = \{50, \},
\textsolidus = \{50, \},
\textsolidus = \{50, \},
\textsolidus = \{50, \},
\textsolidus = \{50, \}.
Now to the math symbols for Computer Modern Roman. Definitions have been extracted from \fontmath.tex. 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 \texttt\texttt and \texttt\textbf alphabets) is OT1/cm, which we’ve already set up above. It’s declared as:

\begin{verbatim}
15.8.6 Computer Modern math

\verbatiminput{fontmath.ltx}
\end{verbatim}
configuration files: Character protrusion

\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 \texttt{letters} (also used as \texttt{mathnormal}) is declared as:

\DeclareSymbolFont{letters} {OML}{cmm}{m}{it}
\SetSymbolFont{letters}{bold}{OML}{cmm}{b}{it}
7013 \texttt{cmr}\texttt{math-letters}
7014 \texttt{cmr-math-letters}
7015 \texttt{cmr-math-letters}
7016 \texttt{cmr-math-letters}
7017 \texttt{cmr-math-letters}
7018 \texttt{cmr-math-letters}
7019 \texttt{cmr-math-letters}
7020 \texttt{cmr-math-letters}
7021 \texttt{cmr-math-letters}
7022 \texttt{cmr-math-letters}
7023 \texttt{cmr-math-letters}
7024 \texttt{cmr-math-letters}
7025 \texttt{cmr-math-letters}
7026 \texttt{cmr-math-letters}
7027 \texttt{cmr-math-letters}
7028 \texttt{cmr-math-letters}
7029 \texttt{cmr-math-letters}
7030 \texttt{cmr-math-letters}
7031 \texttt{cmr-math-letters}
7032 \texttt{cmr-math-letters}
7033 \texttt{cmr-math-letters}
7034 \texttt{cmr-math-letters}
7035 \texttt{cmr-math-letters}
7036 \texttt{cmr-math-letters}
7037 \texttt{cmr-math-letters}
7038 \texttt{cmr-math-letters}
7039 \texttt{cmr-math-letters}
7040 \texttt{cmr-math-letters}
7041 \texttt{cmr-math-letters}
7042 \texttt{cmr-math-letters}
7043 \texttt{cmr-math-letters}
7044 \texttt{cmr-math-letters}
7045 \texttt{cmr-math-letters}
7046 \texttt{cmr-math-letters}
7047 \texttt{cmr-math-letters}
7048 \texttt{cmr-math-letters}
7049 \texttt{cmr-math-letters}
7050 \texttt{cmr-math-letters}
7051 \texttt{cmr-math-letters}
7052 \texttt{cmr-math-letters}
7053 \texttt{cmr-math-letters}
7054 \texttt{cmr-math-letters}
7055 \texttt{cmr-math-letters}
7056 \texttt{cmr-math-letters}
7057 \texttt{cmr-math-letters}
7058 \texttt{cmr-math-letters}
7059 \texttt{cmr-math-letters}
7060 \texttt{cmr-math-letters}
7061 \texttt{cmr-math-letters}
7062 \texttt{cmr-math-letters}
7063 \texttt{cmr-math-letters}
7064 \texttt{cmr-math-letters}
7065 \texttt{cmr-math-letters}
7066 \texttt{cmr-math-letters}
7067  "18 = { , 50}, % \xi
7068  "19 = { 50,100}, % \pi
7069  "1A = { 50, 50}, % \rholn
7070  "1B = { ,150}, % \upsilon
7071  "1C = { 50,100}, % \chi
7072  "1D = { 50, 50}, % \psi
7073  "20 = { 50, 50}, % \omega
7074  "21 = { , 50}, % \varepsilon
7075  "22 = { , 50}, % \vartheta
7076  "23 = { , 50}, % \varpi
7077  "24 = { ,150}, % \varphi
7078  "25 = {100, }, % \varrho
7079  "26 = {100,100}, % \varsigma
7080  "27 = { , 50}, % \varkappa
7081  "28 = {100,100}, % \leftharpoonup
7082  "29 = {100,100}, % \leftharpoondown
7083  "2A = {100,100}, % \rightharpoonup
7084  "2B = {100,100}, % \rightharpoondown
7085  "2C = {300,200}, % \triangleright
7086  "2D = {200,300}, % \triangleleft
7087  "2E = { ,100}, % \lhook
7088  "2F = {200,200}, % \rhook
7089  "3A = { ,500}, % ., \ldotp
7090  "3B = { ,500}, % ,
7091  "3C = {200,100}, % <
7092  "3D = {300,400}, % /
7093  "3E = {100,200}, % >
7094  "3F = {200,200}, % \star
7095  "4B = {100,100}, % \flat
7096  "5E = {200,200}, % \smile
7097  "5F = {200,200}, % \frown
7098  "7C = {100, }, % \jmath
7099  "7D = { ,100} % \wp
7100  }
7101 }
7102 
7103 Math font ‘symbols’ (also used for the \mathcal alphabet) is declared as:
7104 \\
7105 \DeclareSymbolFont{symbols}{OMS}{cmsy}{m}{n}
7106 \SetSymbolFont{symbols}{bold}{OMS}{cmsy}{b}{n}
7107 
7108 { name = cmr-math-symbols }
7109 { encoding = OMS, family = cmsy, series = (m,b), shape = n }
7110 
7111 A = {150, 50}, % \mathcal
7112 C = { ,100},
7113 D = { ,50},
7114 F = { 50,150},
7115 I = { ,100},
7116 J = {100,150},
7117 K = { ,100},
7118 L = {100, },
7119 M = { 50, 50},
7120 N = { 50,100},
7121 P = { , 50},
7122 Q = { 50, },
7123 R = { , 50},
7124 T = { 50,150},
7125 V = { 50, 50},
7126 
7127 Remaining slots in the source file.
7128 
7129 \setprotrusion
7130 { name = cmr-math-symbols }
7131 { encoding = OMS, family = cmsy, series = (m,b), shape = n }
7132 
7133 A = {150, 50}, % \mathcal
7134 C = { ,100},
7135 D = { ,50},
7136 F = { 50,150},
7137 I = { ,100},
7138 J = {100,150},
7139 K = { ,100},
7140 L = {100, },
7141 M = { 50, 50},
7142 N = { 50,100},
7143 P = { , 50},
7144 Q = { 50, },
7145 R = { , 50},
7146 T = { 50,150},
7147 V = { 50, 50},
7148 
7149 
7150 

W = \{ 0 , 50 \},
X = \{100,100\},
Y = \{100, \},
Z = \{100,150\},
\"00 = \{300,300\}, % -
\"01 = \{ 0,700\}, \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\}, \succ
\"18 = \{200,200\}, \triangleright
\"19 = \{150,150\}, \ll
\"1A = \{200,100\}, \gg
\"1B = \{100,300\}, \llless
\"1C = \{200,100\}, \ggg
\"1D = \{100,200\}, \ltimes
\"1E = \{200,100\}, \rtimes
\"1F = \{100,300\}, \divides
\"20 = \{100,200\}, \lll
\"21 = \{200,100\}, \ggg
\"22 = \{100,100\}, \wedge
\"23 = \{100,100\}, \vee
\"24 = \{100,100\}, \uparrow
\"25 = \{100,100\}, \downarrow
\"26 = \{100,100\}, \leftarrow
\"27 = \{100,100\}, \rightarrow
\"28 = \{100,100\}, \uparrow
\"29 = \{100,100\}, \downarrow
\"2A = \{100,100\}, \leftarrow
\"2B = \{100,100\}, \rightarrow
\"2C = \{100,100\}, \leftarrow
\"2D = \{100,100\}, \rightarrow
\"2E = \{100,100\}, \leftarrow
\"2F = \{100,100\}, \rightarrow
\"30 = \{100,100\}, \leftarrow
\"31 = \{100,100\}, \rightarrow
\"32 = \{100,100\}, \leftarrow
\"33 = \{100,100\}, \rightarrow
\"34 = \{100,100\}, \leftarrow
\"35 = \{100,100\}, \rightarrow
\"36 = \{100,100\}, \leftarrow
\"37 = \{100,100\}, \rightarrow
\"38 = \{100,100\}, \leftarrow
\"39 = \{100,100\}, \rightarrow
\"3A = \{200,100\}, \leftarrow
\"3B = \{200,100\}, \leftarrow
\"3C = \{200,100\}, \leftarrow
\"3D = \{200,100\}, \leftarrow
\"3E = \{200,100\}, \leftarrow
\"3F = \{200,100\}, \leftarrow
\"40 = \{100,100\}, \leftarrow
\"41 = \{100,100\}, \leftarrow
\"42 = \{100,100\}, \leftarrow
\"43 = \{100,100\}, \leftarrow
\"44 = \{100,100\}, \leftarrow
\"45 = \{100,100\}, \leftarrow
\"46 = \{100,100\}, \leftarrow
\"47 = \{100,100\}, \leftarrow
\"48 = \{100,100\}, \leftarrow
\"49 = \{100,100\}, \leftarrow
\"4A = \{100,100\}, \leftarrow
\"4B = \{100,100\}, \leftarrow
\"4C = \{100,100\}, \leftarrow
\"4D = \{100,100\}, \leftarrow
\"4E = \{100,100\}, \leftarrow
\"4F = \{100,100\}, \leftarrow
\"50 = \{100,100\}, \leftarrow
\"51 = \{100,100\}, \leftarrow
\"52 = \{100,100\}, \leftarrow
\"53 = \{100,100\}, \leftarrow
\"54 = \{100,100\}, \leftarrow
\"55 = \{100,100\}, \leftarrow
\"56 = \{100,100\}, \leftarrow
\"57 = \{100,100\}, \leftarrow
\"58 = \{100,100\}, \leftarrow
\"59 = \{100,100\}, \leftarrow
\"5A = \{100,100\}, \leftarrow
\"5B = \{100,100\}, \leftarrow
\"5C = \{100,100\}, \leftarrow
\"5D = \{100,100\}, \leftarrow
\"5E = \{100,100\}, \leftarrow
\"5F = \{100,100\}, \leftarrow
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:

```latex
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
```

\section*{AMS symbols}

Settings for the AMS math fonts (amssymb).

Symbol font `a`.

```latex
\SetProtrusion
\SetProtrusion [ name = AMS-a ]
\SetProtrusion [ encoding = U, family = msa ]
```
<table>
<thead>
<tr>
<th>\text{Character Code}</th>
<th>\text{Character}</th>
<th>\text{Description}</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>{200,200}</td>
<td>\upharpoonleft</td>
</tr>
<tr>
<td>19</td>
<td>{200,200}</td>
<td>\downharpoonleft</td>
</tr>
<tr>
<td>1A</td>
<td>{80,100}</td>
<td>\rightarrowtail</td>
</tr>
<tr>
<td>1B</td>
<td>{50,50}</td>
<td>\leftarrowtail</td>
</tr>
<tr>
<td>1C</td>
<td>{80,100}</td>
<td>\rightarrowtail</td>
</tr>
<tr>
<td>1D</td>
<td>{50,80}</td>
<td>\leftharpoons</td>
</tr>
<tr>
<td>1E</td>
<td>{250,}</td>
<td>\lsh</td>
</tr>
<tr>
<td>1F</td>
<td>{,250}</td>
<td>\rsh</td>
</tr>
<tr>
<td>20</td>
<td>{100,100}</td>
<td>\rightsquigarrow</td>
</tr>
<tr>
<td>21</td>
<td>{100,100}</td>
<td>\looparrowleft</td>
</tr>
<tr>
<td>22</td>
<td>{100,50}</td>
<td>\looparrowright</td>
</tr>
<tr>
<td>23</td>
<td>{50,100}</td>
<td>\doteqdot</td>
</tr>
<tr>
<td>24</td>
<td>{50,80}</td>
<td>\circeq</td>
</tr>
<tr>
<td>25</td>
<td>{50,50}</td>
<td>\lapprox</td>
</tr>
<tr>
<td>26</td>
<td>{100,}</td>
<td>\gtrsim</td>
</tr>
<tr>
<td>27</td>
<td>{,100}</td>
<td>\succsim</td>
</tr>
<tr>
<td>28</td>
<td>{50,50}</td>
<td>\simapprox</td>
</tr>
<tr>
<td>29</td>
<td>{100,150}</td>
<td>\dashrightarrow</td>
</tr>
<tr>
<td>30</td>
<td>{100,150}</td>
<td>\dashleftarrow</td>
</tr>
<tr>
<td>31</td>
<td>{50,50}</td>
<td>\equiv</td>
</tr>
<tr>
<td>32</td>
<td>{100,50}</td>
<td>\ensim</td>
</tr>
<tr>
<td>33</td>
<td>{100,100}</td>
<td>\iff</td>
</tr>
<tr>
<td>34</td>
<td>{100,50}</td>
<td>\triangleleft</td>
</tr>
<tr>
<td>35</td>
<td>{50,70}</td>
<td>\bigstar</td>
</tr>
<tr>
<td>36</td>
<td>{50,}</td>
<td>\backprime</td>
</tr>
<tr>
<td>37</td>
<td>{50,50}</td>
<td>\backthreesim</td>
</tr>
<tr>
<td>38</td>
<td>{50,50}</td>
<td>\leftthreetimes</td>
</tr>
<tr>
<td>39</td>
<td>{50,150}</td>
<td>\rightthreetimes</td>
</tr>
<tr>
<td>40</td>
<td>{50,150}</td>
<td>\sideset</td>
</tr>
<tr>
<td>41</td>
<td>{,50}</td>
<td>\circle</td>
</tr>
<tr>
<td>42</td>
<td>{,50}</td>
<td>\circle</td>
</tr>
<tr>
<td>43</td>
<td>{150,}</td>
<td>\lefttriangleleft</td>
</tr>
<tr>
<td>44</td>
<td>{,100}</td>
<td>\righttriangleleft</td>
</tr>
<tr>
<td>45</td>
<td>{100,}</td>
<td>\lefttriangleright</td>
</tr>
<tr>
<td>46</td>
<td>{100,}</td>
<td>\righttriangleright</td>
</tr>
<tr>
<td>47</td>
<td>{,50}</td>
<td>\checkmark</td>
</tr>
<tr>
<td>48</td>
<td>{50,50}</td>
<td>\supset</td>
</tr>
<tr>
<td>49</td>
<td>{50,50}</td>
<td>\subset</td>
</tr>
<tr>
<td>50</td>
<td>{50,}</td>
<td>\smiley</td>
</tr>
<tr>
<td>51</td>
<td>{150,}</td>
<td>\blacktriangleleft</td>
</tr>
<tr>
<td>52</td>
<td>{150,}</td>
<td>\blacktriangleright</td>
</tr>
<tr>
<td>53</td>
<td>{50,50}</td>
<td>\blacktriangleleft</td>
</tr>
<tr>
<td>54</td>
<td>{,50}</td>
<td>\blacktriangleright</td>
</tr>
<tr>
<td>55</td>
<td>{50,50}</td>
<td>\blacktriangleleft</td>
</tr>
<tr>
<td>56</td>
<td>{,50}</td>
<td>\blacktriangleright</td>
</tr>
<tr>
<td>57</td>
<td>{100,}</td>
<td>\triangleleft</td>
</tr>
<tr>
<td>58</td>
<td>{100,}</td>
<td>\triangleleft</td>
</tr>
<tr>
<td>59</td>
<td>{,50}</td>
<td>\triangleleft</td>
</tr>
<tr>
<td>60</td>
<td>{50,50}</td>
<td>\triangleleft</td>
</tr>
</tbody>
</table>
CONFIGURATION FILES : Character protrusion

\[
\begin{align*}
\text{7305} & \quad "6E = \{100,\}, \% \\ulcorner \\
\text{7306} & \quad "6F = \{100\}, \% \\gvertneqq \\
\text{7307} & \quad "70 = \{50,100\}, \% \\ulcorner \\
\text{7308} & \quad "71 = \{100\}, \% \\ulcorner \\
\text{7309} & \quad "75 = \{150,200\}, \% \\dotplus \\
\text{7310} & \quad "76 = \{50,100\}, \% \\backsim \\
\text{7311} & \quad "78 = \{100\}, \% \\urcorner \\
\text{7312} & \quad "79 = \{100,50\}, \% \\urcorner \\
\text{7313} & \quad "7C = \{100,100\}, \% \\intercal \\
\text{7314} & \quad "7D = \{50,50\}, \% \\circledcirc \\
\text{7315} & \quad "7E = \{50,50\}, \% \\circledast \\
\text{7316} & \quad "7F = \{50,50\}, \% \\circleddash
\end{align*}
\]

Remaining slots in the source file.

\[
\begin{align*}
\text{7317} & \quad \}
\text{7318} & \quad \}
\text{7319} & \quad }\msa
\text{7320} & \quad \} \msb
\text{7321} & \quad \SetProtrusion
\text{7322} & \quad [ \text{name} = \text{AMS-b} ]
\text{7323} & \quad \{ \text{encoding} = \text{U}, \text{family} = \text{msb} \}
\text{7324} & \quad \}
\text{7325} & \quad \}
\text{7326} & \quad \}
\text{7327} & \quad \}
\text{7328} & \quad \}
\text{7329} & \quad \}
\text{7330} & \quad \}
\text{7331} & \quad \}
\text{7332} & \quad \}
\text{7333} & \quad \}
\text{7334} & \quad \}
\text{7335} & \quad \}
\text{7336} & \quad \}
\text{7337} & \quad \}
\text{7338} & \quad \}
\text{7339} & \quad \}
\text{7340} & \quad \}
\text{7341} & \quad \}
\text{7342} & \quad \}
\text{7343} & \quad \}
\text{7344} & \quad \}
\text{7345} & \quad \}
\text{7346} & \quad \}
\text{7347} & \quad \}
\text{7348} & \quad \}
\text{7349} & \quad \}
\text{7350} & \quad \}
\text{7351} & \quad \}
\text{7352} & \quad \}
\text{7353} & \quad \}
\text{7354} & \quad \}
\text{7355} & \quad \}
\text{7356} & \quad \}
\text{7357} & \quad \}
\text{7358} & \quad \}
\text{7359} & \quad \}
\text{7360} & \quad \}
\text{7361} & \quad \}
\text{7362} & \quad \}
\text{7363} & \quad \}
\text{7364} & \quad \}
\text{7365} & \quad \}
\end{align*}
\]

Symbol font 'b'.

\[
\begin{align*}
\text{7366} & \quad \}\msb
\text{7367} & \quad \}\msb
\text{7368} & \quad \}\msb
\text{7369} & \quad \}\msb
\text{7370} & \quad \}\msb
\text{7371} & \quad \}\msb
\text{7372} & \quad \}\msb
\text{7373} & \quad \}\msb
\text{7374} & \quad \}\msb
\text{7375} & \quad \}\msb
\text{7376} & \quad \}\msb
\text{7377} & \quad \}\msb
\text{7378} & \quad \}\msb
\text{7379} & \quad \}\msb
\text{7380} & \quad \}\msb
\text{7381} & \quad \}\msb
\text{7382} & \quad \}\msb
\text{7383} & \quad \}\msb
\text{7384} & \quad \}\msb
\text{7385} & \quad \}\msb
\text{7386} & \quad \}\msb
\text{7387} & \quad \}\msb
\text{7388} & \quad \}\msb
\text{7389} & \quad \}\msb
\text{7390} & \quad \}\msb
\text{7391} & \quad \}\msb
\text{7392} & \quad \}\msb
\text{7393} & \quad \}\msb
\text{7394} & \quad \}\msb
\text{7395} & \quad \}\msb
\text{7396} & \quad \}\msb
\text{7397} & \quad \}\msb
\text{7398} & \quad \}\msb
\text{7399} & \quad \}\msb
\text{7400} & \quad \}\msb
\text{7401} & \quad \}\msb
\text{7402} & \quad \}\msb
\text{7403} & \quad \}\msb
\text{7404} & \quad \}\msb
\text{7405} & \quad \}\msb
\text{7406} & \quad \}\msb
\text{7407} & \quad \}\msb
\text{7408} & \quad \}\msb
\text{7409} & \quad \}\msb
\text{7410} & \quad \}\msb
\text{7411} & \quad \}\msb
\text{7412} & \quad \}\msb
\text{7413} & \quad \}\msb
\text{7414} & \quad \}\msb
\text{7415} & \quad \}\msb
\text{7416} & \quad \}\msb
\text{7417} & \quad \}\msb
\text{7418} & \quad \}\msb
\text{7419} & \quad \}\msb
\text{7420} & \quad \}\msb
\text{7421} & \quad \}\msb
\text{7422} & \quad \}\msb
\text{7423} & \quad \}\msb
\text{7424} & \quad \}\msb
\text{7425} & \quad \}\msb
\text{7426} & \quad \}\msb
\text{7427} & \quad \}\msb
\text{7428} & \quad \}\msb
\text{7429} & \quad \}\msb
\text{7430} & \quad \}\msb
\text{7431} & \quad \}\msb
\text{7432} & \quad \}\msb
\text{7433} & \quad \}\msb
\text{7434} & \quad \}\msb
\text{7435} & \quad \}\msb
\text{7436} & \quad \}\msb
\text{7437} & \quad \}\msb
\text{7438} & \quad \}\msb
\text{7439} & \quad \}\msb
\text{7440} & \quad \}\msb
\text{7441} & \quad \}\msb
\text{7442} & \quad \}\msb
\text{7443} & \quad \}\msb
\text{7444} & \quad \}\msb
\text{7445} & \quad \}\msb
\text{7446} & \quad \}\msb
\text{7447} & \quad \}\msb
\text{7448} & \quad \}\msb
\text{7449} & \quad \}\msb
\text{7450} & \quad \}\msb
\text{7451} & \quad \}\msb
\text{7452} & \quad \}\msb
\text{7453} & \quad \}\msb
\text{7454} & \quad \}\msb
\text{7455} & \quad \}\msb
\text{7456} & \quad \}\msb
\text{7457} & \quad \}\msb
\text{7458} & \quad \}\msb
\text{7459} & \quad \}\msb
\text{7460} & \quad \}\msb
\text{7461} & \quad \}\msb
\text{7462} & \quad \}\msb
\text{7463} & \quad \}\msb
\text{7464} & \quad \}\msb
\text{7465} & \quad \}\msb
\end{align*}
\]
"1E = (100, 150), \% \diagup
"1F = (100, 150), \% \diagdown
"20 = (100, 50), \% \varsquare
"21 = (50, 100), \% \varupright
"22 = (100, 50), \% \nsubset
"23 = (50, 100), \% \nsupset
"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, ), \% \ntrianglerighteq
"36 = (100, ), \% \ntrianglerighteq
"37 = (, 100), \% \ntrianglerighteq
"38 = (100, 200), \% \nleftarrow
"39 = (100, 200), \% \nrightarrow
"3A = (100, 100), \% \nLeftarrow
"3B = (50, 100), \% \Rightarrow
"3C = (100, 100), \% \Leftrightarrow
"3D = (100, 200), \% \leftrightarrow
"3E = (50, 50), \% \divideontimes
"3F = (50, 50), \% \emptyset
"60 = (200, ), \% \Finv
"61 = (, 50), \% \Game
"62 = (100, 100), \% \leqsim
"63 = (50, ), \% \beth
"64 = (50, ), \% \gimel
"65 = (100, ), \% \dalet
"66 = (200, ), \% \lesssim
"67 = (, 200), \% \gtrdot
"68 = (100, 200), \% \times
"69 = (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), \% \preccurlyeq
"78 = (100, 100), \% \curvearrowleft
"79 = (50, 150), \% \curvedarrowright
"7A = (50, 200), \% \dagger
"7B = (100, 50), \% \varkappa
"7C = (200, ), \% \backepsilon

Remaining slots in the source file.

15.8.8 Euler

Euler Roman font (package euler).

```latex
\SetProtrusion [ name = euler ]
```

Extended by the eulervm package.

```latex
\SetProtrusion [ name = euler-vm, load = euler ]
```
CONFIGURATION FILES: Character protrusion

```
{ "28 = {100,200},
29 = {100,200},
2A = {100,150},
2B = {200,300},
2D = {100,300},
2E = {100,100},
2F = {100, },
3F = {150,150},
5B = {100,100},
5E = {100,100},
5F = {150,100},
80 = {50,100},
81 = {200,250},
82 = {100,200}
}

Euler Script font (eucal).

\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 = {50,100},
    K = {50,50},
    L = {50,150},
    M = {50,150},
    N = {50,50},
    O = {50,50},
    P = {50,100},
    T = {50,100},
    U = {50,50},
    V = {50,100},
    W = {50,100},
    X = {50,50},
    Y = {50,100},
    Z = {50,100},
    "00 = {200,250},
    "18 = {200,200},
    "3A = {200,150},
    "40 = {100,100},
    "5E = {100,100},
    "5F = {100,100},
    "66 = {50,50},
    "67 = {50,100},
    "6E = {200,200}
  }

\SetProtrusion
  [ name = euscript-vm,
    load = euscript ]
  { encoding = U,
    family = zeus }
  {
```

---

Euler Script font (eucal).
Configuration files: Character protrusion

01 = (600, 600),
02 = (200, 200),
03 = (200, 200),
04 = (200, 200),
05 = (150, 150),
06 = (200, 200),
07 = (200, 200),
08 = (100, 100),
09 = (100, 100),
0A = (100, 100),
0B = (100, 100),
0C = (100, 100),
0D = (100, 100),
0E = (150, 150),
0F = (100, 100),
10 = (150, 150),
11 = (100, 100),
12 = (150, 100),
13 = (100, 150),
14 = (150, 100),
15 = (100, 150),
16 = (200, 100),
17 = (100, 200),
18 = (150, 150),
19 = (150, 100),
1A = (100, 150),
1B = (100, 100),
1C = (100, 100),
1D = (100, 100),
1E = (250, 100),
1F = (100, 250),
20 = (150, 200),
21 = (150, 200),
22 = (150, 150),
23 = (150, 150),
24 = (150, 150),
25 = (100, 100),
26 = (100, 100),
27 = (100, 100),
28 = (100, 100),
29 = (100, 150),
2A = (100, 100),
2B = (100, 100),
2C = (100, 100),
2D = (100, 100),
2E = (150, 150),
2F = (150, 150),
30 = (100, 100),
31 = (100, 100),
32 = (100, 100),
33 = (100, 100),
34 = (100, 100),
35 = (100, 100),
36 = (150, 150),
37 = (150, 150),
38 = (150, 150),
39 = (150, 150),
3A = (150, 150),
3B = (150, 150),
3C = (150, 150),
3D = (150, 150),
3E = (150, 150),
3F = (150, 150),
60 = (100, 200),
61 = (200, 100),
62 = (100, 100),
63 = (100, 100),
64 = (100, 100),
65 = (100, 100),
66 = (300, 300),
67 = (100, 100),
68 = (300, 300),
69 = (100, 100),
6A = (100, 100),
6B = (100, 100),
6C = (100, 100),
6D = (100, 100),
6E = (100, 100),
6F = (100, 100),
Euler Fraktur font (eufrak).

\SetProtrusion
\[ \text{name} = \text{mathfrak} \]
\[ \text{encoding} = \text{U}, \]
\[ \text{family} = \text{euf} \]
\{
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, },
"12 = {500,500},
"13 = {500,500},
! = { ,200},
* = {200,200},
+ = {200,250},
- = {200,200},
\{,\} = {300,300},
. = {400,400},
\{=} = {200,200},
; = { ,200},
; = { ,200},
15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym\textsuperscript{23}).

```latex
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
\SetProtrusion
```

15.9 Interword spacing

Default unit is space.

\textsuperscript{23} Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!
These settings are only a first approximation. The following reasoning is from a mail from Ulrich Dirr, who also provided the sample in figure 1. I do not claim to have coped with the task.

'The idea is – analog to the tables for expansion and protrusion – to have tables for optical reduction/expansion of spaces in dependence of the actual character so that the distance between words is optically equal.

When reducing distances the (weighting) order is:

* after commas

( , ) = ( , -500, 500 ),

* in front of capitals which have optical more room on their left side, e.g., 'A', 'J', 'T', 'V', 'W', and 'Y' [this is not yet possible – RS]

* in front of capitals which have circle/oval shapes on their left side, e.g., 'C', 'G', 'O', and 'Q' [ditto – RS]

* after 'r' (because of the bigger optical room on the righthand side)

r = ( , -300, 300 ),

* [before or] after lowercase characters with ascenders

b = ( , -200, 200 ),
d = ( , -200, 200 ),
f = ( , -200, 200 ),
h = ( , -200, 200 ),
k = ( , -200, 200 ),
l = ( , -200, 200 ),
t = ( , -200, 200 ),

* [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., 'v', or 'w'

c = ( , -100, 100 ),
p = ( , -100, 100 ),
v = ( , -100, 100 ),
w = ( , -100, 100 ),
z = ( , -100, 100 ),
x = ( , -100, 100 ),
y = ( , -100, 100 ),
* [before or] after lowercase characters with x-height plus descender without additional optical space

\[
i = \{ , 50, -50 \},
\]
\[
m = \{ , 50, -50 \},
\]
\[
n = \{ , 50, -50 \},
\]
\[
u = \{ , 50, -50 \},
\]

* after colon and semicolon

\[
: = \{ ,200,-200 \},
\]
\[
; = \{ ,200,-200 \},
\]

* after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

\[
. = \{ ,250,-250 \},
\]
\[
! = \{ ,250,-250 \},
\]
\[
? = \{ ,250,-250 \}
\]

The order has to be reversed when enlarging is needed.’

Questions are:

* Is the result really better?
* Is it overdone? (Try with a factor < 1000.)
* Should the first parameter also be used? (Probably.)
* What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTeX bug with spacing in combination with a non-zero \spaceskip (reported by Axel Berger):

\begin{verbatim}
\parfillskip0pt
\rightskip0pt plus 1em
\spaceskip\fontdimen2\font test test\par
\pdfadjustinterwordglue2
\stbscode\font"t=-50
\stbscode\font\t
\bye
\end{verbatim}

Some more characters in T2A.\footnote{Contributed by Karl Karlsson.}
15.9.1 Nonfrenchspacing

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

From the \TeX\book:

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

The ‘extra space’ (\fontdimen7) for Computer Modern Roman is a third of \fontdimen2, i.e., 333.

\SetExtraSpacing
{ name = nonfrench-cmr, load = default, context = nonfrench }
{ encoding = {OT1,T1,LY1,OT4,QX,T5}, family = cmr }

\latex\.ltx has:

```latex
\def\nonfrenchspacing{
\sfcode`\.: 3000
\sfcode`\?: 3000
\sfcode`\!: 3000
\sfcode`\,: 2000
\sfcode`\,: (333,2000,-667),
\sfcode`\,: (333,2000,-667),
\sfcode`\,: (333,2000,-667),
\sfcode`\,: 2000
\sfcode`\,: 1500
\sfcode`\,: ( , 500,-333),
\sfcode`\,: 1250
\sfcode`\,: ( , 250,-200)
}
```

\fontinst, however, which is also used to create the psnfss font metrics, sets \fontdimen7 to 240 by default. Therefore, the fallback settings use this value for the first component.

\SetExtraSpacing
15.10 Additional kerning

Default unit is 1 em.

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

\SetExtraKerning [ name = empty ]
\SetExtraKerning [ name = french-default, context = french, unit = space ]
\SetExtraKerning [ name = french-guillemets, context = french-guillemets, load = french-default, unit = space ]

15.10.1 French

The ratio of \fontdimen2 to \fontdimen6 varies for different fonts, so that either the kerning of the colon (which should be a space, i.e., \fontdimen2) or that of the other punctuation characters (\text{T\eX}'s \thinspace, i.e., one sixth of \fontdimen6) may be inaccurate, depending on which unit we choose (space or 1em). For Times, for example, a thin space would be 665. I don't know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a space. (Wikipedia\textsuperscript{25} claims it should be a quarter of an em, which seems too much to me; then again, it also says that this was a thin space in French typography.)

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdf\text{T\eX}.  

\SetExtraKerning [ name = french-guillemets, context = french-guillemets, load = french-default, unit = space ]

15.10.2 Turkish

\SetExtraKerning
[ name = turkish, context = turkish ]
\SetExtraKerning
[ name = french-guillemets-OT1, context = french-guillemets, load = french-default, unit = space ]
\SetExtraKerning
[ name = turkish, context = turkish ]
{ encoding = {OT1,T1,LY1} }
{ : = {167,}, % = \thinspace 
! = {167,}, 
= = {167,} 
}
16 OpenType configuration files

These are the configuration files for the following OpenType fonts:

- Latin Modern Roman
- Charis SIL
- Palatino Linotype

The settings are typeset in the respective font.

16.1 Character inheritance

OpenType fonts may differ considerably in how complete their arsenal of glyphs is. Therefore, each font family should have their own inheritance settings.

---

This is file \microtype-utf.dtx.
Available at \url{http://software.sil.org/charis}.
These settings have been contributed by Loren B. Davis.
\begin{verbatim}
7908 Z}, % Greek
7909 a = \{\text{à,á,â,ã,ä,å,ā,ă,ą,ǻ,ȁ,ạ,ả,ấ,ầ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ}\},
7910 æ = \{ǽ\},
7911 c = \{ç,ć,ĉ,ċ,č\},
7912 d = \{đ,ḍ,ḏ\},
7913 e = \{è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ẹ,ẻ,ẽ,ế,ề,ể,ễ,ệ\},
7914 f = \{/f_f\},
7915 g = \{ĝ,ğ,ġ,ģ,ǧ,ǵ\},
7916 h = \{ĥ,ħ,ḥ,ḧ,ḫ\},
7917 i = \{ì,í,î,ï,ĩ,ī,ĭ,į,ı,ȉ,ỉ,ị\},
7918 j = \{ĵ\},
7919 k = \{ķ\},
7920 l = \{ĺ,ļ,ł,ḷ,ḹ\}, % ľ, l·
7921 n = \{ñ,ń,ņ,ň,ṅ,ṇ\},
7922 o = \{ò,ó,ô,õ,ö,ø,ō,ŏ,ő,ọ,ơ,ǫ,ǿ,ȍ,ỏ,ố,ồ,ổ,ỗ,ộ,ớ,ờ,ỡ,ở,ợ\},
7923 r = \{ŕ,ŗ,ř,ȑ,ṛ,ṙ,ṝ\},
7924 s = \{ś,ŝ,ş,š,ș,ṣ\},
7925 t = \{ț,ț,ṭ,ṯ,ẗ\}, % ţ
7926 u = \{ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ư,ȕ,ụ,ủ,ứ,ừ,ử,ữ,ự\},
7927 w = \{ŵ,ẁ,ẃ,ẅ\},
7928 y = \{ý,ŷ,ÿ,ỳ,ỵ,ỷ,ỹ\},
7929 z = \{ź,ż,ž,ẓ\},
\end{verbatim}
\[ T = (T, T, T, T, T, T), \]
\[ T(T), \text{ % Cyrillic} \]
\[ U = (U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U, U), \]
\[ V = (V, V), \]
\[ W = (W, W, W, W, W, W), \text{ % Cyrillic} \]
\[ X = (X, X, X, X), \text{ % Cyrillic} \]
\[ Y = (Y, Y, Y, Y, Y, Y), \text{ % Cyrillic} \]
\[ Z = (Z, Z, Z, Z, Z), \text{ % Cyrillic} \]
\[ a = (a, a, a, a, a), \text{ % Cyrillic} \]
\[ a' = (a', a', a', a', a'), \text{ % Cyrillic} \]
\[ \alpha = (\alpha, \alpha, \alpha, \alpha), \text{ % Cyrillic} \]
\[ ß = (ß, ß, ß, ß), \text{ % Cyrillic} \]
\[ ß = (ß, ß, ß, ß), \text{ % Cyrillic} \]
\[ x = (x, x), \text{ % Cyrillic} \]
\[ y = (y, y, y, y, y, y), \text{ % Cyrillic} \]
\[ y = (y, y, y, y), \text{ % Cyrillic} \]
\[ \beta = (\beta, \beta, \beta, \beta), \text{ % Cyrillic} \]
\[ \beta = (\beta, \beta, \beta, \beta), \text{ % Cyrillic} \]
\[ \gamma = (\gamma, \gamma, \gamma, \gamma), \text{ % Cyrillic} \]
\[ \gamma = (\gamma, \gamma, \gamma, \gamma), \text{ % Cyrillic} \]
\[ \delta = (\delta, \delta, \delta, \delta), \text{ % Cyrillic} \]
\[ \delta = (\delta, \delta, \delta, \delta), \text{ % Cyrillic} \]
\[ \epsilon = (\epsilon, \epsilon, \epsilon, \epsilon), \text{ % Cyrillic} \]
\[ \epsilon = (\epsilon, \epsilon, \epsilon, \epsilon), \text{ % Cyrillic} \]
\[ \theta = (\theta, \theta, \theta, \theta), \text{ % Cyrillic} \]
\[ \theta = (\theta, \theta, \theta, \theta), \text{ % Cyrillic} \]
\[ \iota = (\iota, \iota, \iota, \iota), \text{ % Cyrillic} \]
\[ \iota = (\iota, \iota, \iota, \iota), \text{ % Cyrillic} \]
\[ \kappa = (\kappa, \kappa, \kappa, \kappa), \text{ % Cyrillic} \]
\[ \kappa = (\kappa, \kappa, \kappa, \kappa), \text{ % Cyrillic} \]
\[ \lambda = (\lambda, \lambda, \lambda, \lambda), \text{ % Cyrillic} \]
\[ \lambda = (\lambda, \lambda, \lambda, \lambda), \text{ % Cyrillic} \]
\[ \mu = (\mu, \mu, \mu, \mu), \text{ % Cyrillic} \]
\[ \mu = (\mu, \mu, \mu, \mu), \text{ % Cyrillic} \]
\[ \nu = (\nu, \nu, \nu, \nu), \text{ % Cyrillic} \]
\[ \nu = (\nu, \nu, \nu, \nu), \text{ % Cyrillic} \]
\[ \omega = (\omega, \omega, \omega, \omega), \text{ % Cyrillic} \]
\[ \omega = (\omega, \omega, \omega, \omega), \text{ % Cyrillic} \]
\[ \phi = (\phi, \phi, \phi, \phi), \text{ % Cyrillic} \]
\[ \phi = (\phi, \phi, \phi, \phi), \text{ % Cyrillic} \]
\[ \chi = (\chi, \chi, \chi, \chi), \text{ % Cyrillic} \]
\[ \chi = (\chi, \chi, \chi, \chi), \text{ % Cyrillic} \]
\[ \psi = (\psi, \psi, \psi, \psi), \text{ % Cyrillic} \]
\[ \psi = (\psi, \psi, \psi, \psi), \text{ % Cyrillic} \]
\[ \omega = (\omega, \omega, \omega, \omega), \text{ % Cyrillic} \]
\[ \omega = (\omega, \omega, \omega, \omega), \text{ % Cyrillic} \]
\[ \rho = (\rho, \rho, \rho, \rho), \text{ % Cyrillic} \]
\[ \rho = (\rho, \rho, \rho, \rho), \text{ % Cyrillic} \]
\[ \sigma = (\sigma, \sigma, \sigma, \sigma), \text{ % Cyrillic} \]
\[ \sigma = (\sigma, \sigma, \sigma, \sigma), \text{ % Cyrillic} \]
\[ \tau = (\tau, \tau, \tau, \tau), \text{ % Cyrillic} \]
\[ \tau = (\tau, \tau, \tau, \tau), \text{ % Cyrillic} \]
\[ \upsilon = (\upsilon, \upsilon, \upsilon, \upsilon), \text{ % Cyrillic} \]
\[ \upsilon = (\upsilon, \upsilon, \upsilon, \upsilon), \text{ % Cyrillic} \]
\[ \xi = (\xi, \xi, \xi, \xi), \text{ % Cyrillic} \]
\[ \xi = (\xi, \xi, \xi, \xi), \text{ % Cyrillic} \]
\[ \zeta = (\zeta, \zeta, \zeta, \zeta), \text{ % Cyrillic} \]
\[ \zeta = (\zeta, \zeta, \zeta, \zeta), \text{ % Cyrillic} \]
Unfortunately, I don’t have a Palatino variant containing all of the following glyphs. The settings are typeset in \TeX\ Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as ‘\textordfeminine\textordfeminine’. To see the real settings, consult \texttt{mt-PalatinoLinotype.cfg}.

```latex
\DeclareCharacterInheritance
\{ encoding = \{EU1,EU2,TU\},
\family = \{PalatinoLinotype\} \}
```

Unfortunately, I don’t have a Palatino variant containing all of the following glyphs. The settings are typeset in \TeX\ Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as ‘\textordfeminine\textordfeminine’. To see the real settings, consult \texttt{mt-PalatinoLinotype.cfg}.

```latex
\DeclareCharacterInheritance
\{ encoding = \{EU1,EU2,TU\},
\family = \{PalatinoLinotype\} \}
```

Unfortunately, I don’t have a Palatino variant containing all of the following glyphs. The settings are typeset in \TeX\ Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as ‘\textordfeminine\textordfeminine’. To see the real settings, consult \texttt{mt-PalatinoLinotype.cfg}.

```latex
\DeclareCharacterInheritance
\{ encoding = \{EU1,EU2,TU\},
\family = \{PalatinoLinotype\} \}
```
16.2 Character protrusion
OPENTYPE CONFIGURATION FILES: Character protrusion

8152 \{\} = \{ .500 \},
8153 : = \{ .500 \},
8154 ; = \{ .500 \},
8155 ! = \{ .100 \},
8156 ? = \{ .200 \},
8157 \$ = \{ 50, 50 \},
8158 - = \{ 200, 250 \},
8159 \% = \{ 50, 50 \},
8160 * = \{ 300, 300 \},
8161 + = \{ 250, 250 \},
8162 -backslash = \{ \} = \{ 400, 500 \}, \% /hyphen
8163 - = \{ 400, 500 \}, \% /en dash
8164 -- = \{ 300, 200 \}, \% /endash
8165 _ = \{ 200, 200 \}, \% / underscore
8166 / = \{ 200, 300 \},
8167 /backslash = \{ 200, 300 \},
8168 \' = \{ 300, 400 \}, \% /quotesingle
8169 \' = \{ 500, 700 \}, \' = \{ 500, 600 \},
8170 \'' = \{ 500, 300 \}, \'' = \{ 200, 600 \},
8171 \', = \{ 400, 400 \}, \', = \{ 400, 400 \},
8172 . = \{ 300, 400 \}, \cdot = \{ 300, 300 \},
8173 . = \{ 300, 500 \}, \cdot = \{ 300, 500 \},
8174 \( \) = \{ 100, \}, \( \) = \{ 100, \},
8175 \{ \} = \{ 300, \}, \{ \} = \{ 300, \},
8176 \langle = \{ 200, 100 \}, \rangle = \{ 100, 200 \},
8177 /braceleft = \{ 400, 200 \}, /braceright = \{ 200, 400 \},
8178 /angleleft = \{ 400, \}, /angleright = \{ \, 400 \},
8179 \) = \{ 100, 100 \},
8180 \) = \{ 80, 80 \},
8181 \* = \{ 200, 200 \},
8182 \* = \{ 400, 450 \}, \% /period centered
8183 \`C = \{ 80, 50 \},
8184 \circledC = \{ , 50 \},
8185 \& = \{ 400, 400 \},
8186 \% = \{ 100, 200 \},
8187 \Theta = \{ 100, 100 \},
8188 \Phi = \{ 100, 100 \},
8189 \& = \{ 100, 200 \},
8190 \circledast = \{ 100, 200 \},
8191 \# = \{ 200, 250 \},
8192 \$ = \{ 50, 100 \},
8193 \dagger = \{ 50, 100 \},
8194 \ddagger = \{ 50, 100 \},
8195 \& = \{ 300, 300 \},
8196 \& = \{ 150, 200 \},
8197 \& = \{ 150, 250 \},
8198 \& = \{ 150, 250 \},
8199 \& = \{ 100, \},
8200 \& = \{ 100, 100 \},
8201 \& = \{ 50, 50 \},
8202 \& = \{ 30, 80 \},
8203 \& = \{ 50, 50 \},
8204 \& = \{ 50, 80 \},
8205 \% = \{ .180 \}, \% /Gamma
8206 \Delta = \{ 100, 100 \}, \% /Delta
8207 \Theta = \{ 50, 50 \}, \% /Theta
8208 \Delta = \{ 100, 100 \}, \% /Lambda
8209 \% = \{ , \}, \% /Xi
8210 \% = \{ \}, \% /Pi
8211 \% = \{ 50, 50 \}, \% /Sigma
8212 \% = \{ 100, 100 \}, \% /Upsilon
8213 \% = \{ 50, 50 \}, \% /Phi
8214 \% = \{ 50, 50 \}, \% /Psi
8215 \% = \{ 100, 100 \}, \% /Omega
8216 }
\SetProtrusion
[
  name = LMR-it
]
{
  encoding = {EU1,EU2,TU},
  family = Latin Modern Roman,
  shape = {it,sl}
}
{
  A = \{125,100\},
  \AE = \{125,-55\},
  B = \{90,-40\},
  C = \{145,-75\},
  D = \{75,-28\},
  E = \{80,-55\},
  F = \{85,-80\},
  G = \{153,-15\},
  H = \{73,-60\},
  I = \{140,-120\},
  IJ = \{140,-80\},
  J = \{135,-80\},
  K = \{70,-30\},
  L = \{87,40\},
  M = \{67,45\},
  N = \{75,-55\},
  O = \{150,-30\},
  \OE = \{150,-55\},
  P = \{82,50\},
  Q = \{150,-30\},
  R = \{75,15\},
  S = \{90,65\},
  \S = \{100,-20\},
  T = \{220,-85\},
  U = \{230,-55\},
  V = \{260,-60\},
  W = \{185,-55\},
  X = \{70,-30\},
  Y = \{250,-60\},
  Z = \{90,-60\},
a = \{150,-10\},
b = \{170,\},
c = \{173,-10\},
d = \{150,-55\},
e = \{180,\},
f = \{-,250\},
g = \{150,-10\},
h = \{100,\},
i = \{210,\},
jj = \{210,-40\},
j = \{-,40\},
k = \{110,-50\},
l = \{240,-110\},
m = \{80,\},
n = \{115,\},
o = \{155,\},
q = \{170,-40\},
r = \{155,-40\},
s = \{130,\},
t = \{230,-10\},
u = \{120,\},
v = \{140,-25\},
w = \{98,-20\},
x = \{65,-40\},
y = \{130,-20\},
z = \{110,-80\},
\B = \{170,-85\},
l = \{230,110\},
OPENTYPE CONFIGURATION FILES: Character protrusion

2 = {130,-70}, 3 = {140,-70}, 4 = {130,80}, 5 = {160, }, 6 = {130,-40}, 9 = {155,-80}, . = { .500}, :, = { .300}, & = {130,30}, \% = {180,50}, * = {180,200}, % /hyphen = {400,170}, % /emdash = {100,200}, % /underscore = {200,300}, † = {200, 80}, ‡ = {120, 80}, • = {220,100}, · = {550,300}, ℃ = {170, }, ₡ = {100, 50}, ⊟ = {200, }, ⊝ = {500,300}, ™ = {200, 70}, ° = {500,300}, ¹ = {140,100}, ² = {140,100}, ³ = {140,100}, ¼ = {100,150}, ½ = {250, 80}, ¾ = {250, 80}, ⅓ = {250, 80}, ⅔ = {250, 80}, ⅓ = {250, 80}, ± = {250, 80}, ± = {300,200}, = {150,170}, = {200,200}, = {200,200}, € = {150, }, /one.oldstyle = {100,100}, /two.oldstyle = {100, 80}, /three.oldstyle = {80, 50}, /four.oldstyle = {80, 80}, /five.oldstyle = {50, }, /six.oldstyle = {50, }, /seven.oldstyle = {80, 80}, /eight.oldstyle = {50, },
\[
\Gamma = \{100, 120\}, \% /\Gamma \\
\Delta = \{120, 100\}, \% /\Delta \\
\Theta = \{120, 50\}, \% /\Theta \\
\Lambda = \{130, 100\}, \% /\Lambda \\
\Xi = \{100\}, \% /\Xi \\
\Pi = \{100\}, \% /\Pi \\
\Sigma = \{100, 50\}, \% /\Sigma \\
\Upsilon = \{180, 100\}, \% /\Upsilon \\
\Phi = \{130, 70\}, \% /\Phi \\
\Psi = \{130, 50\}, \% /\Psi \\
\Omega = \{50\}, \% /\Omega 
\]

\[
\langle \text{LatinModernRoman} \rangle \\
\langle \*CharisSIL \rangle \\
\SetProtrusion [ \text{name = Charis-default} ] \\
\{ \text{encoding = \{EU1,EU2,TU\}} \\
\text{family = Charis SIL} \} \\
\{
\ A = (50,50), \\
\ AE = (50,50), \\
\ C = (50,1), \\
\ D = (.50), \\
\ F = (.50), \\
\ G = (50,1), \\
\ J = (100,1), \\
\ K = (.50), \\
\ L = (.50), \\
\ O = (50,50), \\
\ OE = (50,1), \\
\ P = (.50), \\
\ Q = (50,70), \\
\ R = (.50), \\
\ SS = (.40), \% capital sharp s \\
\ T = (50,50), \\
\ V = (50,50), \\
\ W = (50,50), \\
\ X = (50,50), \\
\ Y = (50,50), \\
\ K = (.50), \\
\ L = (.150), \\
\ R = (.50), \\
\ T = (.50), \\
\ V = (50,50), \\
\ W = (50,50), \\
\ X = (50,50), \\
\ Y = (50,50), \\
\ y = (.50), \\
\ 1 = (150,150), \\
\ 2 = (50,50), \\
\ 3 = (50,1), \\
\ 4 = (100,50), \\
\ 6 = (50,1), \\
\ 7 = (50,80), \\
\ 9 = (50,50), \\
\ . = (.600), \\
\ (.) = (.500), \\
\ : = (.400), \\
\ ; = (.300), \\
\ ! = (.100), \\
\ ? = (.200), \\
\ @ = (50,50), \\
\ \sim = (200,250), \\
\ \% = (.50), \\
\ * = (300,300), 
\]
OPENTYPE CONFIGURATION FILES: Character protrusion

+ = (200,250),
/ = (.200),
\backslash = (150,200),
| = (200,200),
- = (400,500), % hyphen
-- = (200,300), % endash
--- = (200,200), % horizontal bar = \texttwelveudash
-- = (150,150), % figure dash = \textthreequartersemdash
_ = (100,100),
{=} = (100,100),
‘ = (300,400), ‘ = (300,400),
“ = (300,300), ” = (300,300),
, = (400,400), . = (300,300),
\iota = (400,300), \iota = (300,400),
\alpha = (200,200), \alpha = (150,300),
\iota = (100, ), \iota = (100, ),
\varepsilon = (200, ), \varepsilon = (100,200),
\epsilon = (200,200), \epsilon = (100,200),
\braceleft = (200, ), /braceright = {,300),
\dagger = (80,80),
\ddagger = (100,100),
\ast = (200,200),
\star = (150,200),
\mu = (100,150),
\epsilon = (50, ),
\circledcirc = (50,100),
\circledcirc = (100,100),
\circledcirc = (100,200),
\circledcirc = (200,200),
\circledcirc = (200,50),
\mu = (100),
\eta = (100),
\theta = (300,400),
\theta = (200,300),
\theta = (100,200),
\theta = (110,200),
\theta = (100,200),
\theta = (100, ),
\theta = (150,200),
\theta = (200,200),
\theta = (250,250),
\theta = (200,200),
\theta = (200,200),
% Cyrillic
\beta = (.50),
\Gamma = (.130),
\chi = (50,50),
\delta = (30,50),
\lambda = (50, ),
\mu = (50,50),
\phi = (50,50),
\psi = (100, ),
\theta = (50, ),
\theta = (50, ),
\epsilon = (50,100),
\epsilon = (50, ),
\phi = (50,50),
\epsilon = (50, ),
\theta = (50,50),
\epsilon = (50, ),
\phi = (50,50),
\epsilon = (50, )
\SetProtrusion

[ name = Charis-it ]

( encoding = (EU1,EU2,TU),

family = Charis SIL,

shape = {it,sl} )

C = (50, ),

G = (50, )
8542 \ J = (50, ), \\
8543 \ L = (50,50), \\
8544 \ O = (50, ), \\
8545 \ Œ = (50, ), \\
8546 \ Q = (50, ), \\
8547 \ S = (50, ), \\
8548 \ Š = (50, ), \\
8549 \ T = (70, ), \\
8550 \ o = (50,50), \\
8551 \ p = ( .50), \\
8552 \ q = (50, ), \\
8553 \ t = ( .50), \\
8554 \ w = ( .50), \\
8555 \ y = ( .50), \\
8556 \ l = (150,100), \\
8557 \ z = (50, ), \\
8558 \ 4 = (100, ), \\
8559 \ 6 = (50, ), \\
8560 \ 7 = (100, ), \\
8561 \ . = ( .700), \\
8562 \ (,)= ( .600), \\
8563 \ : = ( .400), \\
8564 \ ; = ( .400), \\
8565 \ ? = ( .150), \\
8566 \ & = ( .80), \\
8567 \ \%/ = (50,50), \\
8568 \ * = (300,200), \\
8569 \ + = (250,250), \\
8570 \ @ = (80,50), \\
8571 \ \~ = (150,150), \\
8572 \ / = ( .150), \\
8573 \ \//backslash = (150,150), \\
8574 \ \-\% = (300,400), \% hyphen \\
8575 \ \- = (200,300), \% endash \\
8576 \ \--- = (150,200), \% emdash \\
8577 \ . = ( .100), \\
8578 \ \( ) = (200,200), \\
8579 \ \\pm = (150,200), \\
8580 \ \\times = (250,250), \\
8581 \ \\div = (250,250), \\
8582 \ \* = (150,200), \\
8583 \ \cdot = (300,400), \\
8584 \ \' = (400,200), \' = (400,200), \\
8585 \ \" = (300,200), \" = (400,200), \\
8586 \ \, = (200,500), \, = (150,500), \\
8587 \ \( = (300,400), \) = (200,500), \\
8588 \ \& = (200,300), \& = (150,400), \\
8589 \ \( = (200, ), \) = ( .200), \\
8590 \ \< = (200,200), \> = (200,200), \\
8591 \ /bracketleft = (300, ), /bracketright = ( .200), \\
8592 \ \% Cyrillic \\
8593 \ Ж = (50,30), \\
8594 \ Л = (50, ), \\
8595 \ У = (50,30), \\
8596 \ Ф = (50, ), \\
8597 \ Ъ = (100, ), \\
8598 \ ъ = ( .50), \\
8599 \ й = ( .50), \\
8600 \ Э = (50,50), \\
8601 \ Я = (50, ), \\
8602 \ В = (50,50), \\
8603 \ ъ = (50,50), \\
8604 \ й = (140,100), \\
8605 \ й = (70,50), \\
8606 \ й = (50,60),
The small caps glyph names in Charis SIL have changed with version 5.0 of the font. We try to get the names right both with Lua\TeX{} (where we can simply query the font version) and with \TeX{} (where we check for glyph name).

\begin{verbatim}
\% glyph names have changed with version 5.0 of Charis SIL:
\% before: /a.SC, /b.SC, ...
\% after: /a.sc, /b.sc, ...

\ifx\MT@lua\@undefined
  \gdef\MT@get@CHARIS@SC{
    % test whether glyph "a.sc" exists
    \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
      \gdef\MT@CHARIS@SC{sc}%
    \else
      \gdef\MT@CHARIS@SC{SC}%
    \fi
  }
\else
  \gdef\MT@get@CHARIS@SC{
    \gdef\MT@CHARIS@SC{\MT@lua{
      % check font version
      % -- why doesn't this work?:
      \if\font.getfont(\font.current());
        \font.info().filename;\xmpl{\relax > 0
          \gdef\MT@CHARIS@SC{sc}%
        \else
          \gdef\MT@CHARIS@SC{SC}%
        \fi
    \end{verbatim}
\SetProtrusion
\[ name = Charis-sc,
\]
\load = Charis-default,
\command = {MT@get@CHARIS@SC} ]
\encoding = {EU1,EU2,TU},
\family = Charis SIL,
\shape = {sc} }

{% \textsc{A}={100,100}, % etc., doesn't work with \textsc{ }
\}/a.\MT@CHARIS@SC = (100,100),
\}/c.\MT@CHARIS@SC = {50, },
\}/d.\MT@CHARIS@SC = { ,50},
\}/f.\MT@CHARIS@SC = { ,50},
\}/g.\MT@CHARIS@SC = {50, },
\}/j.\MT@CHARIS@SC = (100, ),
\}/k.\MT@CHARIS@SC = ( ,50),
\}/l.\MT@CHARIS@SC = ( ,50),
\}/m.\MT@CHARIS@SC = ( ,50),
\}/o.\MT@CHARIS@SC = {50,50},
\}/p.\MT@CHARIS@SC = {50, },
\}/q.\MT@CHARIS@SC = {50,70},
\}/r.\MT@CHARIS@SC = { ,50},
\}/t.\MT@CHARIS@SC = (50,50),
\}/v.\MT@CHARIS@SC = (50,50),
\}/w.\MT@CHARIS@SC = (50,50),
\}/x.\MT@CHARIS@SC = (50,50),
\}/y.\MT@CHARIS@SC = (50,50)
}

\(\langle \text{CharisSIL} \rangle\)

\(\langle \text{PalatinoLinotype} \rangle\)

\SetProtrusion
\[ name = palatino-default ]
\encoding = {EU1,EU2,TU},
\family = {PalatinoLinotype} ]

\{ A = [50,50],
\} D = [ ,50],
\} J = [50, ],
\} K = [1,50],
\} L = [ ,50],
\} O = [25, ],
\} T = [50,50],
\} V = [50,50],
\} W = [50,50],
\} X = [50,50],
\} Y = [50,50],
\} b = [ ,25],
\} d = [25,30],
\} f = [ ,50],
\} g = [ ,100],
\} k = [ ,50],
\} p = [ ,50],
\} q = [50, ],
\} r = [ ,50],
\} t = [ ,50], \textbullet = [ ,50], \textbullet = [ ,50],
\} v = [75,50],
\} w = [50,50],
\} x = [50,50],
\} y = [50,70],
\} l = [100,50],
OPENTYPE CONFIGURATION FILES: Character protrusion

8731  2 = {25,50},
8732  4 = {50, },
8733  6 = {50, },
8734  9 = {25, },
8735  Α = {100, },
8736  Ω = {25, },
8737  ., = {,700}, ..., = {,350}, ..., = {,150},
8738  :) = {500},
8739  ; = {500},
8740  − = {200,250},
8741  & = {50,100},
8742  \% = {100,100},
8743  » = {200,200},
8744  ÷ = {250,250},
8745  ( = {100, }, ) = {,300},
8746  / = {200,300},
8747  ñ = {400,500},
8748  \textendash = {300,300}, \textemdash = {200,200},
8749  \textquoteleft = {500,700}, \textquoteright = {500,700},
8750  \textquotedblleft = {300,400}, \textquotedblright = {300,400},
8751  \textbackslash = {200,300},
8752  \quotesinglbase = {400,400}, \quotedblbase = {400,400},
8753  \guilsinglleft = {400,400}, \guilsinglright = {300,500},
8754  \textexclamdown = {100, }, \textquestiondown = {100, },
8755  \textbraceleft = {400,200}, \textbraceright = {200,400},
8756  \textless = {200,100}, \textgreater = {100,200},
8757  ≤ = {200,100}, ≥ = {100,200},
8758  \textminus = {300,300},
8759  \texttrademark = {200,200},
8760  \textcopyright = {200,200},
8761  \textregistered = {200,200},
8762  \textdegree = {300,300},
8763  ¦ = {450,500}, ¬ = {250,150},
8764  \textsuperscript{0} = {150,250},
8765  † = {200,250}, ‡ = {200,250},
8766  π = {50, },
8767  ¶ = {,50},
8768  № = {100,150},
8769  \textcopyright = {100,200},
8770  ‑ = {400,500}, \textemdash = {200,200},
8771  ‒ = {200,200},
8772  \textasciicircum = {200,200},
8773  \% = {50,50},
8774  % /a.sc = {50,50},
8775  \textdegree = {300,300}, \textendash = {300,300},
8776  \textasciicircum = {100,0},
8777  \textasciicircum = {100,0},
8778  \textasciicircum = {300,300}, \textasciicircum = {300,300},
8779  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8780  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8781  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8782  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8783  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8784  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8785  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8786  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8787  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8788  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8789  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8790  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8791  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8792  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8793  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8794  \textasciicircum = {200,400}, \textasciicircum = {200,400},
8795  }
\SetProtrusion

[ name = palatino-it ]

( encoding = \{EU1,EU2,TU\},
family = \{PalatinoLinotype\},
shape = \{it,sl\} )

{ A={50,50},
Æ = {50, },
B = {50, },
C = {50, },
D = {50,50},
E = {50, },
F = {50, },
G = {50, },
H = {50, },
K = {50, },
L = {50, },
O = {50, },
Œ = {50, },
P = {50, },
Q = {50, },
R = {50, },
S = {50, },
$ = {50, },
T = {100, },
U = {50, },
V = {100,50},
W = {50, },
X = {50, },
Y = {100,50},
b = { ,50},
c = {25, },
g = {75, },
i = {25, },
m = { ,50},
n = { ,50},
p = { ,25},
q = {25, },
x = { ,50},
1 = {100, },
2 = {50, },
4 = {50, },
7 = {50, },
.. = { ,500}, \ldots = { ,350}, \ldots = { ,200},
\ldots = { ,500},
:\ldots = { ,300},
? = { ,300}, \ldots = { ,300},
\& = {50,50},\%/ = {100,100},
* = {200,200},
+ = {150,200},
\@ = {50,50},
- = {200,150},
( = {200, }, ) = { ,200},
/ = {100,200},
\textendash = {300,300}, \textemdash = {200,200},
\textquoteleft = {700,400}, \textquoteright = {700,400},
\textquotedblleft = {500,300}, \textquotedblright = {500,300},
\_ = {100,100},
\textbackslash = {100,200},
\quotesinglbase = {500,500}, \quotedblbase = {400,400},
\guilsinglleft = {400,400}, \guilsinglright = {300,500},
\SetProtrusion

[ name = palatino-sc, load = palatino-default ]

( PalatinoLinotype )
17 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.
The following displays the current font stretched by 5\%, normal, and shrunk by 5\%:

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net.
A The title logo

This is microtype-logo.dtx. You may treat this file in three different ways:
• compile it by itself
• \input it in the body of a dtx file
• \input it in the preamble: it then provides the command \printlogo, which will do just that

The first two cases require the style file microtype-doc.sty, which can be generated from microtype.ins with:

```
\makefile{microtype-doc.sty}{docsty}
```

Here’s how the logo on the title page was created. It has nothing to do with microtype, actually, but uses fontinst. It is based on an experiment I posted to the de.comp.text.tex newsgroup. It will show:
• the character
• the T\TeX box
• the bounding box
• kerns

A.1 Macros

To run this file, T\TeX needs to find the \texttt{afm} file (either in the \texttt{TEXINPUTS} path, or in the current working directory).

First input fontinst.

```
\input fontinst.sty
```

\texttt{bbox.sty} is an addition to \texttt{fontinst}, which makes dimensions of the bounding boxes available (and was written by Hàn Thê Thành, by the way). These dimensions are specified in the \texttt{afm} file, but not used by T\TeX, which is why fontinst will discard them otherwise.

```
\input bbox.sty
```

\texttt{\tempdim} Allocate some dimen registers.

```
\newdimen\tempdim
\newdimen\fboxrulei
\newdimen\kernboxheight
```

\texttt{\fboxrulei} Frame width of the box as T\TeX sees it.

```
\fboxrulei=0.1pt
```

\texttt{\fboxruleii} Frame width of the bounding box.

```
\fboxruleii=0.1pt
```

\texttt{\kernboxheight} Height of the box indicating the kern.

```
\kernboxheight=5pt
```

\texttt{\scaletoem} An auxiliary macro. Return a dimension relative to the \texttt{em}-width of the font. Requires e-T\TeX.

```
\setcommand\scaletoem#1{\dimexpr #1 sp*\fontdimen6\font/1000\relax}
```

\texttt{\showlogo} A \texttt{fontinst} incantation whose sole purpose is to produce the logo. Its argument is a string (letters only).

```
\fontinstcc
\def\showlogo#1{\%
Some fonts do not specify the \texttt{fontdimen} (width of an \texttt{em}) in the \texttt{afm} file. In this case, use the font size, which is correct in most cases.
\ifdim\fontdimen6\font = 0pt
\Typeout{ ***\texttt{Warning: no \texttt{fontdimen}}-\texttt{6} specified---\texttt{\fboxrulei} would be 0 pt}\
\texttt{\fboxrulei}=\texttt{the font size}\n\fi
\def\showlogo#1{\%
```
\installfonts
```

29 Note that the logo module will not be created when installing microtype. Instead, the source file microtype-logo.dtx is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the \texttt{pdftk} tool.

30 Message ID: 42aa36b75052436639b4e6d930newsread2.arcor-online.net
Layers.

```
\makeatletter
\def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
\ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
\ifx\mt@order\@undefined\let\mt@order\@empty\fi
\xdef\mt@order{\mt@order[(Logo]}
\let\mtl@resources\@empty
\def\mtl@register#1{\immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
\expandafter\xdef\csname mtl@#1\endcsname{\the\pdflastobj\space 0 R }
\xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
\xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
\xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
\mtl@register{canvas}
\mtl@register{characters}
\mtl@register{bounding-boxes}
\mtl@register{TeX-boxes}
\xdef\mt@order{\mt@order]}
\global\let\mtl@objects\mt@objects
\def\togglelayer#1#2{\pdffontlink width \wd\logobox height \ht\logobox depth \dp\logobox
user{/Subtype/Link}
/BS << /Type/Border/W 0 >> /H/O
/A << /S/SetOCGState /State[/Toggle \csname mtl@#1\endcsname] >>}
\setcommand\printbbs#1{\setbox0\hbox{#1} \leavevmode
\kern-\fboxrulei
\getboundarychars#1
\tempdim=\dimexpr\wd0 - (\scaletoem{\lpcode\font\firstchar} + \\
\scaletoem{\rpcode\font\lastchar})\relax
\kern\dimexpr\scaletoem{\lpcode\font\firstchar}\relax
\lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor} \hrule width \tempdim
height \dimexpr\dp0+\ht0+0.15em\relax} \kern-\tempdim
\getlastchar
\printbbss #1\relax\relax}
\getboundarychars Get first . . . and last character.
\getlastchar Get first . . . and last character.
```
\def\getlastchar#1#2{\ifx\relax#2\relax\def\lastchar{`#1}\else\expandafter\getlastchar\fi #2\relax}

\printbbss Loop over all characters of the string.
\def\printbbss#1#2#3\relax{\ifx\relax#1\relax\else\ifx\relax#2\relax\printbb{#1}{}\else\printbb{#1}{#2}\fi\expandafter\printbbss\fi #2#3\relax}

\printbb Record the kern between the current and the following character, then print the character. \kerning is a fontinst command.
\setcommand\printbb#1#2{\setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}\showboxes{#1}}

This could be another application.
\setcommand\showboxes#1{\leavevmode\color{texcolor}We have to record the width of the glyph.\setbox0\hbox{{\color{textcolor}#1}}\global\tempdim=\wd0\relax1. The \TeX\ box: Print a frame in color texcolor. This frame shows the glyph as \TeX\ sees it.
\mtl@layer{TeX-boxes}{\hbox{\lower\dimexpr \dp0 + \fboxrulei\relax\hrule height\fboxrulei\hbox{\hrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax\phantom{\unhcopy0}}\hrule width\fboxrulei\hbox{\hrule height\fboxrulei}}}2. The character: Now we step back and print the actual glyph. We hold it back until now, so that it will be printed on top of its box.
\kern=\wd0\mtl@layer{characters}{\hbox{\box0}}Step back by the amount that the character's bounding box differs from the \TeX\ box on the left side.
\kern=\dimexpr\scaletoem{\bbleft{#1}}-\tempdim-\fboxruleii\relax
3. The bounding box: will be printed in color bbcolor.

\begin{verbatim}
\lower\dimexpr-\scaletoem{\bbbottom{#1}}+\fboxruleii\relax
\hbox{\hrule height\fboxruleii}
\hbox to \dimexpr\scaletoem{\numexpr\bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii{\vrule height \dimexpr\scaletoem{\numexpr\bbtop{#1}-\bbbottom{#1}\relax}+\fboxruleii}
\hrule height\fboxruleii
\vfill
\hrule width\fboxruleii
\hrule height\fboxruleii
\kern-\fboxruleii
\end{verbatim}

4. The kern: We also print a small box in color kerncolor indicating the kerning between the current and the next character; filled for negative kerns, empty for positive kerns.

\begin{verbatim}
\kern\scaletoem{\width{#1}-\bbright{#1}\relax}
\iftexnum\thekern<0\color{kerncolor}\kern\scaletoem{\thekern}\lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoem{\thekern}\relax height \kernboxheight\hbox{\hrule width\fboxruleii}}\hrule height\fboxruleii}{\color{texcolor}\iftexnum\thekern=0 \else\lower\kernboxheight\hbox{\vbox{\hrule height\fboxrulei}}\hrule width\fboxrulei\hrule height\fboxrulei}{\vrule width\fboxrulei}\kern\scaletoem{\thekern-2\fboxrulei}\vrule width\fboxrulei}}\hrule height\fboxrulei}}\vfill
\end{verbatim}

\newbox\logobox
\def\printlogo{\setbox\logobox=\hbox{\vbox{\MakePercentComment
This is the Kepler MM font used in the logo.
\def\logofont{pkpri9e10}\transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}\font\thelogofont=\logofont\space at 82pt
This would load the italic Palatino font instead.
\def\logofont{pplri}\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}\edef\logofont{\logofont8r}\font\thelogofont=\logofont\space at 78pt
Load the font.
Protrusion values (overdone for didactic reasons).

Now we can generate the logo.

```
\pdfliteral direct{/SXS gs}\
\showlogo{Microtype}\
% \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%\
% \kern5pt\[3\baselineskip%\
% \long\def\@makefntext##1{%\
% \leftskip 0pt\
% \parindent 0pt\
% \everypar{\parindent 0pt}%\
% \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}%\
% \footnotetext[1]{This graphic display on a %\togglelayer{canvas}{canvas} the \togglelayer{characters}{characters}, %\togglelayer{bounding-boxes}{bounding boxes} %and \togglelayer{TeX-boxes}{\TeX boxes}.}\
\edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}\
\immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}\
\immediate\pdfxform attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}\
\immediate pdfformattr {/Properties <</mtl@resources>> /ExtGState << /SXS \the\pdflastobj\space 0 R >> }\
\pdfannot\logodimens{%/Subtype/Widget /FT/Btn /T(Logo) /F 4 % why did I say this? /AP << /N \the\pdflastxform\space 0 R >> /AA << /E << /S/SetOGState /State[/Toggle mtl@characters] >> /X << /S/SetOGState /State[/Toggle mtl@characters] >> /D << /S/SetOGState /State[/Toggle \csname mtl@characters\endcsname] >> /U << /S/SetOGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >> >> }\
\vspace{3\baselineskip}
}
```

Our font.

```
\pdfmapline{+pkpmmri8r10 KeplMM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
```

Define colours (thered and thegreen are copied from microtype.dtx).

```
\def\mtdefinecolors{
\definecolor{thered}{rgb}{0.65,0.04,0.07}
\definecolor{thegreen}{rgb}{0.06,0.44,0.08}
\colorlet{pxcolor}{thegreen!50} % Tex boxes
\colorlet{kerncolor}{pxcolor} % negative kerns
\colorlet{bbcolor}{thered!50} % bounding box
\colorlet{bccolor}{black18} % canvas
\colorlet{bicolor}{black!50} % baseline
\colorlet{textcolor}{black!40} % text
}
```

Use with microtype.dtx.

```
\ifx\documentclass@twoclasseserror
\usepackage[xcdraw]{xcolor}
\mtdefinecolors
\else
```
A.2 Document

Now we can start the document.

```
\documentclass[10pt,a4paper]{ltxdoc}
\providecommand\MakePercentComment{relax}
\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
```

Re-use the preamble from `microtype.dtx`.

```
\usepackage{microtype-doc}
\usepackage{attachfile}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
\makeatother
\begin{document}
You are currently reading this.
\DocInput{microtype-logo.dtx}
\newpage
And here it is:
\vfill
\begin{center}
\printlogo \null
\end{center}
\vfill
\expandafter\enddocument
\fi
That's it.
```

B The letterspacing illustration

This is `microtype-lssample.dtx`. You may treat this file in three different ways:

- compile it by itself
- \input it in the body of a dtx file
- \input it in the preamble: it then provides the commands
  - \lssample: prints the letterspacing illustration
  - \anchorarrow: anchors an arrow for layer (#1)
  - \showarrow: toggles layer (#1) or (#2), and prints (#2)

The first two cases require the style file `microtype-doc.sty`, which can be generated from `microtype.ins` with:

```
\makefile{microtype-doc.sty}{docsty}
```

```
\ifx\lssample\undefined
\input{lssample}
```

Upon popular request, here’s how I’ve created the letterspacing illustration.\footnote{Note that the \lssample module will not be created when installing microtype. Instead, the source file `microtype-lssample.dtx` is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the pdftk tool.}

B.1 Macros

Rule width and image height and depth.

```
\makeatletter
\newdimen\lsamount
\newdimen\lsrule
\lsrule=0.2pt
\lsamount=8pt
\def\lsheight{8pt}
\def\lsdepth{12pt}
```
Our font (Adobe Caslon).

\begin{verbatim}
\def\ls{}\fontfamily{paca}\selectfont
\def\lsfont{\fontfamily{paca}\selectfont}
\def\dolss#1#2\enddols{\ifx\empty#2\empty\divide\lsamount 2\fi}\ls{#1}
\def\ls#1{\begin{tikzpicture}[remember picture,line width=\lsrule]
\tikzstyle{every node}=[inner sep=0pt]
\mts@layer{stuff}{{
\node[draw=thegrey,fill=theshade,outer sep=\lsrule,anchor=base,font=\lsfont](\phantom{#1});
}}\end{tikzpicture}}
\def\lssp#1#2#3#4{\begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
\mts@layer{stuff}{{
\coordinate(#1space) at (#2/2,\lsdepth/2);
\coordinate(#1stretch) at (#2+#3/2,+0pt);
\coordinate(#1shrink) at (#2-#4/2,+0pt);
\draw[thegreen,fill=thegreen!50,use as bounding box](0,0) rectangle ++(#2,\lsdepth);
\draw[thegreen,fill=thegreen!30](#2,-\lsrule) rectangle ++ (#3,-4pt+\lsrule);
\draw[thegreen,fill=thegreen!50](#2,-\lsrule) rectangle ++(-#4,-4pt-\lsrule);
\draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
}}\end{tikzpicture}}
\end{verbatim}

One \texttt{tikz} picture for each letter.

\begin{verbatim}
\def\ls#1{\begin{tikzpicture}[remember picture,line width=\lsrule]
\tikzstyle{every node}=[inner sep=0pt]
\mts@layer{stuff}{{
\node[draw=thegrey,fill=theshade,outer sep=\lsrule,anchor=base,font=\lsfont](\phantom{#1});
}}\end{tikzpicture}}
\end{verbatim}

The bounding box.

\begin{verbatim}
\def\ls#1{\begin{tikzpicture}[remember picture,line width=\lsrule]
\tikzstyle{every node}=[inner sep=0pt]
\mts@layer{stuff}{{
\node[draw=thegrey,fill=theshade,outer sep=\lsrule,anchor=base,font=\lsfont](\phantom{#1});
}}\end{tikzpicture}}
\end{verbatim}

The letter.

Two auxiliary coordinates.

\begin{verbatim}
\def\lssp#1#2#3#4{\begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
\mts@layer{stuff}{{
\coordinate(#1space) at (#2/2,\lsdepth/2);
\coordinate(#1stretch) at (#2+#3/2,+0pt);
\coordinate(#1shrink) at (#2-#4/2,+0pt);
\draw[thegreen,fill=thegreen!50,use as bounding box](0,0) rectangle ++(#2,\lsdepth);
\draw[thegreen,fill=thegreen!30](#2,-\lsrule) rectangle ++ (#3,-4pt+\lsrule);
\draw[thegreen,fill=thegreen!50](#2,-\lsrule) rectangle ++(-#4,-4pt-\lsrule);
\draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
}}\end{tikzpicture}}
\end{verbatim}

Draw the interword space.

\begin{verbatim}
\def\lssp#1#2#3#4{\begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
\mts@layer{stuff}{{
\coordinate(#1space) at (#2/2,\lsdepth/2);
\coordinate(#1stretch) at (#2+#3/2,+0pt);
\coordinate(#1shrink) at (#2-#4/2,+0pt);
\draw[thegreen,fill=thegreen!50,use as bounding box](0,0) rectangle ++(#2,\lsdepth);
\draw[thegreen,fill=thegreen!30](#2,-\lsrule) rectangle ++ (#3,-4pt+\lsrule);
\draw[thegreen,fill=thegreen!50](#2,-\lsrule) rectangle ++(-#4,-4pt-\lsrule);
\draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
}}\end{tikzpicture}}
\end{verbatim}
Layers.
\def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
\def\mtsx@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
\ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
\ifx\mt@order \@undefined\let\mt@order \@empty\fi
\xdef\mt@order{\mt@order[(Sheep)\]}
\let\mts@resources\@empty
\def\mts@register#1{%
\immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
\expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj 0 R }
\xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
\xdef\mt@order{\mt@order\csname mts@#1\endcsname}
\xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
\mts@register{stuff}
\mts@register{tracking}
\mts@register{ispace}
\mts@register{ospace}
\mts@register{istretch}
\mts@register{ishrink}
\mts@register{okern}
\mts@register{ligature}
\mts@register{_compatibility}
\xdef\mt@order{\mt@order]}

Anchor point for the arrow in the code.
\newcommand\anchorarrow[1]{%
\tikz[remember picture,overlay]\node(#1_c){};}

Add an arrow from code to image.
\newcommand\add@arrow[5][left]{%
\tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
\mtsx@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
}%

Toggle layer.
\def\toggle@layer#1#2#3{%
\pdfstartlink
\user{/Subtype/Link /BS << /Type/Border/W 0 >> /H/O}
\xdef\toggle@layer{#2}%
\pdfendlink}

\newcommand\showarrow[2][]{%
\ifx\relax\relax\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi%
\toggle@layer{\@tempa}{#2}}
The environment for our illustration.

\def\ls@sample#1{{\parskip 4pt \parindent 0pt \\
\vskip4pt 
\leftskip 15pt
\mt@pseudo@marg{Click on the image to show the kerns and spacings involved. Click on emphasised words in the text below to reveal the relation of image and code.}}
\mt@layer{_compatibility}{{\Ifyouhad a \acronym{PDF} viewer that understands \acronym{PDF}, \smaller{1.5}, you could hide the arrows selectively.}}
\vskip\mt@unvdimen}
\vskip-4pt
\setlength\fboxsep{4pt}
\leavevmode
\pdfstartlink
\user{/Subtype/Link/BS << /Type/Border/W 0 >> /H/O /A << /SetOCGState /State[/Toggle \mts@stuff] >>}
\fcolorbox{theframe}{theshade}{\fontsize{34}{38}\selectfont #1}
\pdfendlink
\par\medskip
}\edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
}

Now define the illustration to be used in the document.

\def\ls@sample{\ls@sample{\dols{0pt}{Stop}\lssp{o}{0.45em}{0.25em}{0.15em}\dols{0.16em}{{st}ealing}\hskip-\dimexpr 0.08em+\lsrule\relax\lssp{i}{13.82pt}{4.65pt}{2.08pt}\dols{0.16em}{sheep}\dols{0pt}{!}}}

Don't forget to add the arrows.

\vspace{-\baselineskip}
\add@arrow{red} {tracking}{lsamount \textunderscore c.east}{a\_ls}
\add@arrow{red} {okern} {okernend \textunderscore c.east}{p\_ls}
\add@arrow{green} {ospace} {ospace \textunderscore c.east} {ospace}
\add@arrow{green} {ispace} {ispace \textunderscore c.center} {ispace}
\add@arrow{green!75} {istretch}{istretch_c.east}{istretch.north}
\add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
\add@arrow{green!75} {ostretch}{ostretch_c.east} {ostretch.north}
\add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
\add@arrow[right]{grey}{ligature}{nolig_c.east} {st.center}

This is for use with \microtype\dtx
\if\documentclass[10pt,a4paper]{ltexdoc}
\usepackage{tikz}
\else
\fi

B.2 Document
Re-use the preamble from microtype.dtx.

\usepackage{microtype-doc}
\usepackage{attachfile}
\usepackage{tikz}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects]
\makeatother
\begin{document}

You are currently reading this.
\DocInput{microtype-lssample.dtx}

Now show what we are able to do.
\begin{verbatim}
SetTracking
\[ no ligatures = {"\anchor{nolig}f},
spacing = {60"\anchor{ispace}0 *="%00, "\anchor{ishrink}"00*},
outer spacing = {4"\anchor{ospace}50,"%2\anchor{ostretch}50,1\anchor{oshrink}50},
outer kerning = {"\anchor{okernbegin}"*,"%2\anchor{okernend}"*} ]
{ encoding = * }
{ 1"\anchor{lsamount}60 }
\end{verbatim}
and then write:
\begin{verbatim}
Stop \textls{stealing sheep}!
\end{verbatim}
this is the (typographically dubious) outcome:
\lssample

While the word `Stop' is not letterspaced, the space between the letters in
the other two words is expanded by the \showarrow{tracking}{tracking-amount}{red}
of 160/1000\,em, \allowbreak\,0.16\,em.
The \showarrow{ispace}{inner-space}{green} within the letterspaced text is
increased by 60\%, while its \showarrow{istretch}{stretch}{green} amount is
decreased by 10\% and the \showarrow{ishrink}{shrink}{green} amount is left
untouched.
The \showarrow{ospace}{outer-space}{green} (of 0.45\,em) immediately before the
piece of text may \showarrow{ostretch}{stretch}{green} by 0.25\,em and
\showarrow{oshrink}{shrink}{green} by 0.15\,em.
Note that there is no outer space after the text, since the exclamation mark
immediately follows; instead, the default \showarrow{okern}{outer-kern}{red}
of half the letterspace amount (0.08\,em) is added.
Furthermore, one \showarrow{ligature}{grey} wasn't broken up, because we
neglected to specify the `|s|' in the [no ligatures] key.
\expandafter\enddocument
\fi
</lssample>
C Change history

2004/09/11 Version 1.0
General: Initial version

2004/09/21 Version 1.1
General: configuration file names in lowercase (suggested by Harald Harders)
remove 8-bit characters from the configuration files (suggested by Harald Harders)
Protrusion: add factors for some more characters
settings for Adobe Minion (contributed by Harald Harders)
\DeclareCharacterInheritance: new command: possibility to specify character inheritance
\MT@findfile: fix: also check whether the file for the base font family has already been loaded
\MT@getbasefamily: only remove suffixes 'x' or 'j'
\MT@getlistname: don't check for empty attributes

2004/11/03 Version 1.2
Font aliases: declare cmor as an alias of cmr
Font sets: new: allmath and basicmath
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding
add settings for Computer Modern Roman math symbols
\MT@familyalias: define alias font name as an alternative, not as a replacement
\MT@getbasefamily: also remove 'w' (swash capitals)
\MT@get@highlevel: check whether defaults have changed

2004/10/03 Version 1.3
General: fix: specifying load option does no longer require to give a name, too
Font aliases: declare aer, zer and hfor as aliases of cmr

2004/11/22 Version 1.4
General: check for pdfcprot
\Microtypesetup: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options
Protrusion: tweak quote characters for cmr variants

2004/11/03 Version 1.4a
General: new option: final
\MT@cfg@catcodes: fix: reset some more catcodes when reading files (reported by Michael Hoppe)

CHANGE HISTORY 226
CHANGE HISTORY

2004/11/26

**Version 1.4b**

General: fix: set catcodes before reading global configuration file (reported by Christoph Bier) .................. 127
optimisation: use less `expandafter' and \csnames \efcode \csname \ psychologically
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl) .................................................. 150
slanted like italics .................................................. 159
\MT@checklistfamily: fix: don't try alias family name if encoding failed ........................................ 60
\MT@getbasefamily: fix: failed for font names of the form abczz (reported by Georg Verweyen) ............. 87
\MT@get@slot: don't define \@char globally (save stack problem) .................................................. 90
\pdfoutput: don't set \pdfoutput globally (save stack problem) .................................................. 46
\MTsetup@PDF: new message if \pdfoutput is changed .................................................. 132
\MT@use@set: don't use undeclared font sets .................................................. 108

2004/12/15

**Version 1.5**

General: defaults: step: 4 (suggested by Hàn Thế Thành) .................................................. 126
new option: selected, by default false (suggested by Hàn Thế Thành) .................................. 124
Documentation: add 'Short history' .................................................. 30
add note about DVoutput option .................................. 8
Inheritance: remove `{ss` from TI list, add `{NJ ... 144
Protrusion: settings for Bitstream Charter .................................................. 151
\DeclareMicrorfontAlias: remove spaces around arguments .................................. 109
\MT@fpq@catcode: reset catcode of `\string (compatibility with Turkish babel) .................. 87
\MT@fix@catcode: reset catcode of `\string (compatibility with chemsym) .................. 35
\MT@set@ex@codes: don't use \string (ltx encoding) .................................. 65
\MT@set@ex@codes: introduce \string factor .................................. 61
\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters .................................. 61
\MT@setup@expansion: defaults: calculate step as \min(stretch, shrink)/5 .......................... 133
defaults: turn off expansion for DVI output .................................. 133
disable automatic expansion for DVI output .................................. 134

2005/01/24

**Version 1.6**

General: defaults: turn off expansion for old pdfTeX versions .................................. 127
load a font if none is selected .................................. 56
restructure dtx file .................................. 126
test whether `{pickup\font has changed .................................. 100
test whether numeric options receive a number .................................. 126
use e-Tex's `{csname and `{\@d@fined if defined .................................. 44
Protrusion: add italic uppercase Greek letters .................................. 159
improve settings for numbers (pointed out by Peter Muthesius) .................................. 153
tune CMR math letters (OML encoding) .................................. 180
\MT@set@charwd: use e-Tex's `{\fontcharwd, if available .................................. 64
\MT@pin@list: correct message if selected is false .................................. 89
\MT@set@ex@codes: introduce \string factor option .................................. 69
\MT@set@pr@codes: introduce factor option .................................. 61
\MT@setup@expansion: disable automatic expansion for old pdfTeX versions .................................. 134
defaults: turn off expansion for DVI output .................................. 133
disable automatic expansion for DVI output .................................. 134

2005/02/02

**Version 1.6a**

Documentation: add table of fonts with tailored protrusion settings .................................. 21
\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem reported by Bernard Gaulle) .................................. 90
\pdfTeX@no: new macro .................................. 39
\MT@reset@ef@codes: only reset \@efcodes for older pdfTeX versions .................................. 69

2005/03/23

**Version 1.7**

General: allow specification of size ranges (suggested by Andreas Bühmann) .................................. 105
disallow automatic expansion if pdfTeX too old ............ 117
fix: remove space after `autoexpand' .................................. 117
new value for verbosen option: errors .................................. 125
shorter command names .................................. 50
warning when running in draft mode .................................. 131
Documentation: add hint about compatibility .................................. 26
remove table of match order (now table 4 on page 88) .................................. 12
Protrusion: fix: remove \ from OT1, add \textbackslash slash to TI encoding .................................. 154
\LoadMicrotypeFile: new command (suggested by Andreas Bühmann) .................................. 109
\Microtype@Hook: new command for font package authors .................................. 128
\microtypesetup: fix: warning also when setting to (no)compatibility .................................. 128
\MT@begin@catcodes: also use inside configuration commands .................................. 87
Version 1.8

General: \SetProtrusion: new key: unit 116
if font substitution has occurred, set up the substitute font, not the selected one 98
new option: config to load a different main configuration file 127
new option: unit, by default character 126
Documentation: add example for factor option add example of how to get rid of a widow (suggested by Adam Kucharczyk) 15
add hint about error messages 27
Font aliases: declare pxr and txr as aliases of ppl resp. ptm 141
Font sets: add U encoding to all math 140
Inheritance: remove \DJ from T1 list (it's the same as \DJ) 144
Protrusion: add LY1 characters for Times settings for AMS math fonts 159
verified settings for slanted Computer Modern Roman 168
\addaccent: fix: disable micro-typographic setup inside \addaccent (reported by Stephan Hennig) 100
\DeclareMicrotypeAlias: warning when overriding an alias font 109
\DeclareMicrotypeSetDefault: new command: set default font set 108
\MT@cfg@catcodes: reset catcodes of the remaining ASCII characters 87
\MT@check@rlist: made recursive 121
\MT@curr@list@name: new macro: current list type and name 96
\MT@declare@sets: warning when redefining a set 103
\MT@define@set@key: use comma lists instead of token lists 104

2005/06/23

Version 1.9

General: \DeclareMicrotypeSet: new key: font 106
\SetProtrusion: value 'relative' renamed to 'character' for key unit 116
allow context-specific font setup 98
compatibility with \TeX Live hack (reported by Herbert Voß) 38
disable microtype setup inside hyperref's \pdfstringdef (reported by Hån Thål Thanh) 55
fix: use true as the default value for option unit: rename value relative to character 126
Documentation: add hint about verbatim environment 25
add remark about Type 1 fonts required for automatic font expansion 8
Font aliases: declare qpl and qtm (qfonts, \TeX Gyre) as aliases of ppl resp. ptm 141
Font sets: add OT1 encoding to text sets 140
add TS encoding to text sets 140

2005/10/28


### Change History

#### Version 1.9a

- **General:** `\pdfstrcmp` as default list name ........................................ 114
- **new option:** `defersetup`, by default true ..................................... 124
- **remove** superficial test whether `\pickuptextfont` has changed .......... 100
- **Documentation:** add explanation for error message in DVI mode ........ 27
- **add** explanation for error message with non-Type 1 fonts ................ 27
- **Font** aliases: declare `mbch` (mathdesign) as an alias of `bch` ........ 142
- **Protrusion:** `\twospace` from OT1 encoding .................................. 155
- **settings for T5** encoded Charter ............................................. 150
- **\microtypesetup:** inside the preamble, accepts all package options ..... 128
- **\MT@checkfontcx:** optimise context-sensitive setup .................... 101
- **\MT@definefontkey:** don’t expand variables immediately (requested by Georg Verweyen) ......................................................... 104
- **\MT@get@highlevel:** no longer check whether defaults have changed .... 104
- **\MT@defined@font@ot1:** new macro: true case only .......................... 44
- **\MT@fint:** use `\pdfmatch` if available ................................... 45
- **\MT@fstreq:** use `\pdfstricmp` if available .................................. 46
- **\MT@checklist@family:** fix ..................................................... 48
- **\MT@info@missing@char:** info instead of warning (after Michael Hoppe reported that the ‘l’ ligature is missing in Palatino SC) .................................................... 65
- **\MT@is@feature:** new macro: check for `\pdfTeX` feature ................. 51
- **\MT@map@clist@ot1:** following `\pdfTeX` ..................................... 47
- **\MT@permute@0000:** don’t define permutations for unused encodings ... 120
- **\MT@rem@from@clist:** fix ..................................................... 48
- **\MT@setup0:** defer setup until the end of the preamble .................. 51

#### Version 1.9b

- **General:** compatibility with `\listings`: sanitise more catcodes (reported by Holger Uhr) ................................................................. 55
- **compatibility with the extendedchar option of the \listings package** .......... 55
- **Documentation:** activate expansion in the distributed PDF ................ 1
- **add** samples of micro-typographic features .................................. 4
- **\MT@features:** use throughout the package to adjust to beta-ness .......... 51
- **\MT@definedimen:** use `\pdfmatch` if available ................................ 46
- **\MT@warn@code@too@large:** fix calculation with present factor .......... 65

#### Version 1.9c

- **Documentation:** add example of how to increase protrusion of footnote markers (suggested by Georg Verweyen) .................................. 22
- **Protrusion:** settings for URW Garamond ....................................... 151
- **\MT@define@fontkey@\font:** fix: context was ignored .......................... 113
- **\MT@define@code@\font@key@size:** fix: embrace ................................ 113
- **\MT@tempsize in \csname (bug introduced in v1.9b) ** ........................ 113

#### Version 1.9d

- **Font:** `\md` instead of `\n` series in basic sets ............................. 140
- **\add@QXencoding** to text sets .................................................. 140
- **Inheritance:** add list for QX encoding (contributed by Maciej Eder) .... 146
- **Protrusion:** settings for QX encoding (contributed by Maciej Eder) ...... 157
- **settings for Euro symbols (Adobe, ITC, marvosym) ............................. 191
- **\MT@getopt:** optimise: don’t reset when \preset option
CHANGE HISTORY 230

2006/07/28  
Version 1.9e

General: fix: default value for activate: true 123
Documentation: add hint about unknown encodings 26
Font aliases: declare zeur and zeus (eurvm) as aliases of eur resp. eus ( euler ) 130
Inheritance: adapt to marvosym's changed encoding 148
Protrusion: complete settings for Euler Fraktur and Script fonts 180
fix: forgotten comma in mt-mvs.cfg; adapt to marvosym's changed encoding 191

2006/09/09  
Version 1.9f

Protrusion: fix: euler-vm did not load euler settings 187
\@curr@list@name: fix: \MessageBreak must not be expanded 186
\@get@inh@list: new macros: global variants 44
\@get@inh@list: fix: input encoding must be set after the inheritance list has been parsed 90
\@get@list: new macro 43

2007/01/14  
Version 2.0

General: compatibility with listings; set catcode of backlash to zero (reported by Steven Bath) 55
compatibility with soul: register \textls and \lsstyle 55
new option: babel, by default false (language-dependent setup suggested by Ulrich Dirr) 124
new option: letterspace, by default 100 126
new package letterspace: a stripped-down version, containing the letterspacing commands only 1
option 'babel': fix: switch off French babel's short-hands properly (reported by Daniel Flipo) 139
option 'babel': switch off Turkish babel's short-hands 139
option 'unit', \SetProtrusion: deprecate value 'relative' completely 116
Documentation: add hint about how to increase font_max and font_mem_size 28
add hint about warning when tracking and expansion is applied to a font 28
add remark about 'draft' option disabling microuype (noted by Michalis Miatidis) 9
qualify hint about web documents with regard to older pdfTeXX versions 25
qualify hints about expansion error messages with regard to older pdfTeXX versions 27
Font sets: new: footnotesize and scriptsize 140
new: smallcaps 140
\DeclareMicrotypeBabelHook: new command: interaction with babel 110
\lsstyle: fix: font switches don't pose a problem anymore 77
fix: letterspacing commands may be nested 77
new command: letterspacing 77
totally redone, using the new \lsstyle 77
\\@declare@sets: fix: empty size list when redefining set 103
\\@is@symbol: made even more robust 94
\\@load@inputenc: sanitise catcodes before loading input encoding (problem with listings) 67
\\@pdf@text@no: case 6: pdffTExX 1.40 40
\\@setup@noligatures: maybe disable \\@noligatures after the preamble 138
\\@split@name: adjust to possible letterspacing 59
\SetExtraKerning: new command: additional kerning 112
\SetExtraSpacing: new command: adjustment of interword spacing 112
\SetTracking: new command: tracking 111
\\textls: new command: letterspacing 82
starred version: remove spaces around text 82
\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations 37

2006/09/09  
Version 1.9e

General: compatibility with listings; set catcode of backlash to zero (reported by Steven Bath) 55
compatibility with soul: register \textls and \lsstyle 55
new option: babel, by default false (language-dependent setup suggested by Ulrich Dirr) 124
new option: letterspace, by default 100 126
new package letterspace: a stripped-down version, containing the letterspacing commands only 1
option 'babel': fix: switch off French babel's short-hands properly (reported by Daniel Flipo) 139
option 'babel': switch off Turkish babel's short-hands 139
option 'unit', \SetProtrusion: deprecate value 'relative' completely 116
Documentation: add hint about how to increase font_max and font_mem_size 28
add hint about warning when tracking and expansion is applied to a font 28
add remark about 'draft' option disabling microuype (noted by Michalis Miatidis) 9
qualify hint about web documents with regard to older pdfTeXX versions 25
qualify hints about expansion error messages with regard to older pdfTeXX versions 27
Font sets: new: footnotesize and scriptsize 140
new: smallcaps 140
\DeclareMicrotypeBabelHook: new command: interaction with babel 110
\lsstyle: fix: font switches don't pose a problem anymore 77
fix: letterspacing commands may be nested 77
new command: letterspacing 77
totally redone, using the new \lsstyle 77
\\@declare@sets: fix: empty size list when redefining set 103
\\@is@symbol: made even more robust 94
\\@load@inputenc: sanitise catcodes before loading input encoding (problem with listings) 67
\\@pdf@text@no: case 6: pdffTExX 1.40 40
\\@setup@noligatures: maybe disable \\@noligatures after the preamble 138
\\@split@name: adjust to possible letterspacing 59
\SetExtraKerning: new command: additional kerning 112
\SetExtraSpacing: new command: adjustment of interword spacing 112
\SetTracking: new command: tracking 111
\\textls: new command: letterspacing 82
starred version: remove spaces around text 82
\tracingmicrotypeinpdf: new debug method: mark all fonts with PDF annotations 37

2006/09/09  
Version 1.9f

Protrusion: fix: euler-vm did not load euler settings 187
\@curr@list@name: fix: \MessageBreak must not be expanded 186
\@get@inh@list: new macros: global variants 44
\@get@inh@list: fix: input encoding must be set after the inheritance list has been parsed 90
\@get@list: new macro 43
**Version 2.1**

General: compatibility with pinyin: disable microtype in \pymacro (reported by Sven Nau- mann) .......................... 56

fix: letterspace package forgot to load keyval . 41

\srlig: new command: protect ligatures in letter- spaced text ....................... 78

**Version 2.2**

General: disable microtype if wordcount is loaded
(reported by Ross Hetherington) .......... 51

new option: copyfonts .................. 125

simplify key declarations ............... 114

use catcode trickery for e-Tex test .... 38

Documentation: add hint about error message with pdfTeX 1.40 --------------- 27

add hint about extra TOC leader dot (first dis- covered by Morten Høgholm) ...... 25

add overview .............................. 5

logo transparency and amusement ....... 1

Font aliases: declare chr (math) as an alias of bch (reported by Geoff Vallis) 142

declare fp9x, fp9j (FPL Neu) as aliases of pp [x] 142

Font sets: default set for tracking: smallcaps ........................................ 141

Inheritance: remove 'a' to 'd'7 144

Protrusion: settings for Bistream Letter Gothic .................. 151

Spacing: add sample ........................ 192

Tracking: add ligatures that are to be disabled .... 148

\DeclareMicrotypeVariants: new command .......... 109

\DisableLigatures: new optional argument: disable selected ligatures only .... 110

\srlig: always defined ................. 78

\Microchecklist\font: fix: construct font name from characteristics .... 61

\Microcopy\font: optionally work on copies of fonts .......................... 57

\Microget\basefont: redone, working on font names

and suffixes of arbitrary length .......... 87

\Microget\charwd: subtract letterspacing amount from width ............... 64

\Microget\basefont: fix again: remember base font in a macro ............... 78

\Microfdimen: employ Lua\TeX\ features if available ..................... 46

\Microfint: employ Lua\TeX\ features if available ....................... 45

\Microfstret: employ Lua\TeX\ features if available ..................... 46

\fix: \e\TeX\ version shouldn't use \x and \y (found by Wiebke Petersen) .... 46

**Version 2.3**

General: disable \microtypecontext\ in hyperrefs
\pdfstringdef ................................ 55

fix: really switch off Turkish shorthands ...... 139

new value for verbose option: silent (suggested by Karl Berry) ............. 125

turned some warnings into errors .......... 125

Documentation: add kerning sample .......... 18

add letterspacing illustration ........... 17

\do\@substyle\@correction: remember substitute font for all times (reported by Stephan Hennig) .................. 100

\srlig: redone: extract outer kerns from current letterspacing amount .... 78

\Microchecklist\font: redone: use \pdftmatch to

make it bullet-proof ...................... 78

\Microorig\@pickup\font: compatibility with CJK: also check for its definition ........ 98

\textls: fix: use \hmode@\group ............. 82

...
\texttt{\textbackslash MT@set@tr@codes}: also adjust tracking if protrusion is not enabled, and even for letterspace (reported by Stephan Hennig) .......................... 75
\texttt{\textbackslash MT@SetTracking}: sanity check for value .......................... 112
\texttt{\textbackslash MT@setup@tracking}: enable protrusion when tracking is enabled ......................................................... 136
\texttt{\textbackslash MT@tr@outer@r@}: only change pre outer space if it contains shrink ......................................................... 80

2008/02/29

\textbf{Version 2.3a}

General: fix test for soul under plain \TeX .......................... 55
too old for extensions .......................... 137
Documentation: add hint about babel having to be loaded first ......................................................... 26
add table of available and enabled features ...................... 7
mention soulutf8 ........................................ 30
Protrusion: adjust LMR quotation marks again .................. 156
\texttt{\textbackslash MTerror@doesn@t@work}: error messages if pd\TeX is not
\texttt{\textbackslash MT@set@codes}: also adjust tracking if protrusion is not enabled, and even
for letterspace (reported by Stephan Hennig) .......................... 75

2008/06/04

\textbf{Version 2.3b}

\texttt{\textbackslash MT@exp@\textbackslash pt@copy@cs}: new macro: reduce save stack size .......................... 43
\texttt{\textbackslash MT@font@copy@copy}: enable font copies also with protrusion contexts (reported by Nathan Rosenblum) .......................... 58
\texttt{\textbackslash MT@get@size@copy}: grouping ........................................ 106
\texttt{\textbackslash MT@noligatures@\textbackslash r@}: fix: warning messages for unknown slots .......................... 84
\texttt{\textbackslash MT@orig@\textbackslash pick@upfont}: compatibility with CJKutf8: .......................... 81
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 131
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 193
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 267
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 329
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 391
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 453
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 515
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 577
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 639
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 701
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 763
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 825
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 887
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 949
\texttt{\textbackslash MT@tr@outer@l@}: in class options list .......................... 1011

2008/11/11

\textbf{Version 2.3c}

General: \texttt{\textbackslash l@s@style}: disable for \texttt{\textbackslash \textit{La}tex} ........................................ 35
Documentation: add hint about spacing being experimental ......................................................... 26
Inheritance: add \texttt{\textbackslash textcom@b@label@\textbackslash \textit{S}et\textbackslash \textit{S}t\textit{e}} to \texttt{\textit{QX}} en...
\texttt{\textbackslash MT@tr@inner@kerning}: allow empty value for outer kerning ......................................................... 83
\texttt{\textbackslash textls}: make math mode aware ........................................ 82

2009/03/27

\textbf{Version 2.3d}

General: \texttt{\textbackslash pinyin@compatibility \textit{check}: (reported by Silas S. Brown)} .......................... 56
move setup to the very end (for Colin Rouke) ...................... 139
\texttt{\textbackslash MT@\textbackslash in\textbackslash annot}: use \texttt{pdf\textbackslash tex\textbackslash nds} for debugging .......................... 37
\texttt{\lst@style}: disable for \texttt{La}tex ........................................ 77
\texttt{\textbackslash micro\textbackslash type\textbackslash setup}: select font after setup ...................... 128
\texttt{\textbackslash MT@check\textbackslash active\textbackslash set}: warning for missing default sets .......................... 128
\texttt{\textbackslash MT@ua}: update for Lua\TeX \textit{0.36} ........................................ 41
\texttt{\textbackslash MT@\textbackslash set\textbackslash tr@codes}: allow zero tracking ........................................ 74
\texttt{\textbackslash MT@\textbackslash set\textbackslash tr@zero}: fix: allow switching off tracking ........................................ 78

2009/11/09

\textbf{Version 2.3e}

Documentation: suggest to patch \texttt{\textbackslash verbatim} instead of \texttt{\textbackslash verbatim} ........................................ 26
Expansion: settings for T2A encoding (contributed by Karl Karlsson) ........................................ 149
Font sets: sc* instead of sc in smallcaps set ...................... 140
add T2A encoding ........................................ 140
Protrusion: settings for T2A encoding (contributed by Karl Karlsson) ........................................ 157
Spacing: settings for T2A encoding (contributed by Karl Karlsson) ........................................ 193
\texttt{\textbackslash MT@\textbackslash get\textbackslash font\textbackslash dimension\textbackslash x}: fix: gobbling settings with tracking failed (reported by Leo) ........................................ 62
\texttt{\textbackslash MT@setup@kerning}: allow empty value for outer kerning ......................................................... 83
\texttt{\textbackslash \textit{textls}}: make math mode aware ........................................ 82

2009/11/09
**CHANGE HISTORY**

2010/01/10

**Version 2.4**

General: new file `microtype.lua` containing the lua functions (contributed by Elie Roux) .......... 43

Protrusion: settings for T2A encoded Minion (contributed by Karl Karlsson) .......... 157

2013/03/13

**Version 2.5**

General: allow contexts for LuaTeX (fixes) .......... 115

disable \texttt{Output} option for XeLaTeX .......... 124

fix: check whether \texttt{/file}/\texttt{line} list name already exists (reported by Till A. Heilmann) .......... 114

letterspacing with LuaTeX 0.62 .......... 74

new files: \texttt{microtype-pdftex.def}, \texttt{microtype-latex.def}, containing engine-specific definitions .......... 38

protrusion with XeLaTeX .......... 40

restore \texttt{\vspace} inside \texttt{listings} (reported by Rolf Dieterich) .......... 55

Documentation: add hint about LuaTeX compatibility improvements to Computer Modern Roman italics (contributed by Hendrik Vogt) .......... 160

Font aliases: declare \texttt{lmsy} and \texttt{lmr} as aliases of \texttt{cmr} resp. \texttt{cmm} (reported by Jonas Hogstrom) .......... 141

declare \texttt{garamondx} as aliases of \texttt{ugm} and \texttt{Cmm} (reported by Georg Duffner and Rolf Dieterich) .......... 142

Font sets: add EU1 and EU2 encodings .......... 140

Inheritance: add rudimentary list for EU1 and EU2 .......... 147

Protrusion: add default lists for EU1 and EU2 improvements to Computer Modern Roman italics (contributed by Hendrik Vogt) .......... 160

Tracking: add EU2 encoding to default list .......... 148

\texttt{\special{character:inheritance}}: allow more than one encoding .......... 117

\texttt{\DeclareMicrotypeAlias}: ignore spaces (suggested by Hàn Thế Thành) .......... 61

\texttt{\loadmicrotypefile}: remove all spaces in font name .......... 109

\texttt{\style}: fix: ensure to set up math fonts (reported by RazorXr) .......... 78

2013/05/23

**Version 2.5a**

General: use \texttt{latexbase} instead of \texttt{latexextra} (contributed by Elie Roux) .......... 43

Documentation: add notes on typesetting the documenta- .....

\texttt{\loadmicrotypealias}: ignore spaces (suggested by Hàn Thế Thành) .......... 61

\texttt{\loadmicrotypefile}: remove all spaces in font name .......... 109

\texttt{\style}: fix: ensure to set up math fonts (reported by RazorXr) .......... 78

2016/05/01

**Version 2.6**

General: load \texttt{luatex} with LuaTeX .......... 43

```latex
\define@code{key}{family}: compatibility with \texttt{fontspec}: remove its internal counter (reported by Till A. Heilmann) .......... 113
```

Documentation: add note about partial incompatibil- ity with xe\LaTeX{} and \texttt{latexja} (reported by Sam Mason) .......... 27

```latex
\DeclareMicrotypeAlias}: ignore spaces (suggested by Hàn Thế Thành) .......... 61
```

Missing characters printed with Charis SIL .......... 197
### CHANGE HISTORY

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016/05/14</td>
<td>2.6a</td>
<td>General: fixes for letterspace package with LuaTeX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\texttt{\fontspec{font} } fix lua function (reported by Herbert)</td>
</tr>
<tr>
<td>2017/07/07</td>
<td>2.7</td>
<td>General: drop \texttt{luatexbase} with recent \texttt{luatex}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>warning with minimal class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Documentation: add remark about automatic font expansion with \texttt{dvilla}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mention that additional kerning does not work in math mode (discovered by Daniel)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Font aliases: declare aliases for newpx</td>
</tr>
<tr>
<td></td>
<td></td>
<td>declare aliases for newtx</td>
</tr>
<tr>
<td></td>
<td></td>
<td>declare aliases for tempora</td>
</tr>
<tr>
<td></td>
<td></td>
<td>declare aliases for XCharter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>declare Latin Modern Roman as alias of \texttt{lmc} with new \texttt{ft} format (reported by Ulrike Fischer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protrusion: automatically choose correct names for Charis Tiny small caps (reported by \texttt{ltcomdata})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\texttt{\lsstyle{fix: prevent infinite loop with \texttt{psnfss and \texttt{exscale} packages (reported by user11126, solution by Ulrike Fischer})}}</td>
</tr>
<tr>
<td>2018/01/4</td>
<td>2.7a</td>
<td>General: disallow non-automatic expansion with \texttt{luatex}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\texttt{\fontspec{font} } auto: remove \texttt{autoexpand} for \texttt{luatex} 1.0.6 (reported by Ulrike Fischer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with \texttt{luatex}, non-automatic font expansion is no longer possible (as confirmed by Hans Hagen)</td>
</tr>
<tr>
<td>2019/02/28</td>
<td>2.7b</td>
<td>General: update lua function \texttt{microtype.warn} after changes in \texttt{luaotfload} (reported by Moritz Wemheuer and Ulrike Fischer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Documentation: update hint about non-7-bit character\texttt{ers (notified by Frank Mittelbach) }</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inheritance: add \texttt{textquotedblleft} ligature to \texttt{OT4} (reported by Franz Wexler)</td>
</tr>
</tbody>
</table>
|            |           | \texttt{\fontspec{font} } info\texttt{missingchar: fix message for glyphs spe-}
\textbf{INDEX}

\begin{center}
\begin{tabular}{lrr}
\hline
\textbf{Options} & \textbf{Letterspace} & \textbf{8} \\
\hline
\texttt{DVIoutput} & \texttt{letterspace} & \texttt{8}
\end{tabular}
\end{center}

\small
\textit{INDEX} 235

\section*{D Index}

Numbers in upright shape refer to the page where the corresponding entry is described (bold face) resp. occurs. Numbers in italics refer to the code line where the corresponding entry is defined (underlined) resp. used.

\begin{center}
\begin{tabular}{ll}
\hline
\textbf{Options} & \textbf{Letterspace} & \textbf{8} \\
\hline
\texttt{DVIoutput} & \texttt{letterspace} & \texttt{8}
\end{tabular}
\end{center}

\textbf{Commands}

\begin{center}
\begin{tabular}{ll}
\hline
\texttt{\textbackslash DeclareCharacterInheritance} & \texttt{letterspace} & \texttt{8} \\
\hline
\end{tabular}
\end{center}

\begin{itemize}
\item \texttt{\textbackslash DeclareCharacterInheritance} \texttt{20}
\item \texttt{\textbackslash DeclareMicrotypeAlias} \texttt{21}
\item \texttt{\textbackslash DeclareMicrotypeBabelHook} \texttt{23}
\item \texttt{\textbackslash DeclareMicrotypeSet*} \texttt{10}
\item \texttt{\textbackslash DeclareMicrotypeSet} \texttt{10}
\item \texttt{\textbackslash DisableLigatures} \texttt{24}
\item \texttt{\textbackslash DVIoutput} \texttt{8}
\item \texttt{\textbackslash SetTracking} \texttt{15}
\item \texttt{\textbackslash SetExtraSpacing} \texttt{19}
\end{itemize}

\begin{itemize}
\item \texttt{\textbackslash A0poster} \texttt{(package)} \texttt{106}
\item \texttt{\textbackslash addaccent} \texttt{(package)} \texttt{2907}
\item \texttt{\textbackslash adjustspacing} \texttt{(package)} \texttt{4277}
\item \texttt{\textbackslash babel} \texttt{(package)} \texttt{21, 141}
\item \texttt{\textbackslash babel} \texttt{(option)} \texttt{9, 23, 27, 31, 124, 230}
\item \texttt{\textbackslash babel} \texttt{(package)} \texttt{2, 3, 5, 17, 23, 24, 27, 53}
\item \texttt{\textbackslash chcalc} \texttt{(package)} \texttt{227}
\item \texttt{\textbackslash chcalc} \texttt{(option)} \texttt{9, 20, 32, 127, 228}
\item \texttt{\textbackslash chcalc} \texttt{(package)} \texttt{21, 142, 231}
\item \texttt{\textbackslash cm-super} \texttt{(package)} \texttt{8}
\item \texttt{\textbackslash cm-super} \texttt{(option)} \texttt{9, 132}
\item \texttt{\textbackslash combine} \texttt{(package)} \texttt{139, 230}
\item \texttt{\textbackslash cfrak} \texttt{(package)} \texttt{21, 141}
\item \texttt{\textbackslash cfrak} \texttt{(option)} \texttt{9, 20, 32, 127, 228}
\item \texttt{\textbackslash cfrak} \texttt{(package)} \texttt{21, 143, 71, 3310}
\item \texttt{\textbackslash cm-super} \texttt{(package)} \texttt{27, 56, 98, 231, 232, 234}
\item \texttt{\textbackslash cmtex} \texttt{(package)} \texttt{99, 232}
\item \texttt{\textbackslash cm-super} \texttt{(option)} \texttt{57, 124, 125, 231}
\item \texttt{\textbackslash color} \texttt{(package)} \texttt{8}
\item \texttt{\textbackslash color} \texttt{(package)} \texttt{9, 132}
\item \texttt{\textbackslash combine} \texttt{(package)} \texttt{139, 230}
\item \texttt{\textbackslash defersetup} \texttt{(option)} \texttt{9, 20, 32, 127, 228}
\item \texttt{\textbackslash defersetup} \texttt{(package)} \texttt{2844, 2852, 2856, 2866}
\item \texttt{\textbackslash disableligatures} \texttt{(package)} \texttt{24, 40, 62, 2349}
\item \texttt{\textbackslash docstrip} \texttt{(package)} \texttt{44}
\item \texttt{\textbackslash draft} \texttt{(option)} \texttt{9, 124, 125, 230}
\item \texttt{\textbackslash dsfont} \texttt{(package)} \texttt{62}
\item \texttt{\textbackslash defersetup} \texttt{(option)} \texttt{8, 9, 124, 227, 233}
\item \texttt{\textbackslash euler} \texttt{(package)} \texttt{96, 187, 230}
\item \texttt{\textbackslash eulervm} \texttt{(package)} \texttt{21, 142, 187, 230}
\item \texttt{\textbackslash eurotcs} \texttt{(package)} \texttt{191}
\item \texttt{\textbackslash eurosans} \texttt{(package)} \texttt{191}
\item \texttt{\textbackslash expandglyphs} \texttt{(package)} \texttt{1529}
\item \texttt{\textbackslash etoolbox} \texttt{(package)} \texttt{6, 11, 123, 133}
\item \texttt{\textbackslash euscale} \texttt{(package)} \texttt{234}
\item \texttt{\textbackslash eucal} \texttt{(package)} \texttt{188}
\item \texttt{\textbackslash eucal} \texttt{(package)} \texttt{190}
\end{itemize}
E The \LaTeX{} Project Public License

\textit{LPPL Version 1.3c 2008-05-04}

Copyright 1999, 2002–2008 \LaTeX{}3 Project

Preamble

The \LaTeX{} Project Public License (LPPL) is the primary license under which the \LaTeX{} kernel and the base \LaTeX{} packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is \LaTeX{}-related (such as a \LaTeX{} package), but it is written in such a way that you can use it even if your work is unrelated to \TeX{}.

The section ‘WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE’, below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may...
be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the \LaTeX{} Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document `cfgguide.tex' and `modguide.tex' in the base \LaTeX{} distribution for suggestions.

Definitions

In this license document the following terms are used:

Work: Any work being distributed under this License.

Derived Work: Any work that under any applicable law is derived from the Work.

Modification: Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify: To apply any procedure that produces a Derived Work under any applicable law.

Distribution: Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun's Network File System (NFS).

Compiled Work: A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer: A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the 'Copyright Holder' under any applicable law.

Base Interpreter: A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a \LaTeX{}-Format' or in the case of files belonging to the \LaTeX{}-language' a program implementing the \LaTeX{} language'.

Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.

2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.

3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.

4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.

5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.

6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.

   (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.

   (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

   (c) No information in the Derived Work implies that
any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.

(d) You distribute at least one of the following with the Derived Work:

- A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
- Information that is sufficient to obtain a complete, unmodified copy of the Work.

7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.

8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.

9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.

10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.

(b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.

12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.

2. If this search is successful, then enquire whether the Work is still maintained.

(a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
(b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a \TeX{} work, this could be done, for example, by posting to comp.text.tex.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.

(b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.

4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the \LPPL{} license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

### Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

### Choosing This License or Another License

If for any part of your work you want or need to use distribution conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the \LPPL{} or otherwise give the impression that your work is distributed under the \LPPL{}.

The document ‘modguide.tex’ in the base \TeX{} distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing \TeX{} under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to \TeX{}, the discussion in ‘modguide.tex’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

### A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

### How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

`\begin{verbatim}
\% pig.dtx
\% Copyright 2005 M. Y. Name
\%
\% This work may be distributed and/or modified under the
\% conditions of the LaTeX Project Public License, either version 1.3
\% of this license or (at your option) any later version.
\% The latest version of this license is in
\% https://www.latex-project.org/lppl.txt
\% and version 1.3 or later is part of all distributions of LaTeX
\% version 2005/12/01 or later.
\%
\% This work has the \LPPL{} maintenance status 'maintained'.
\% The Current Maintainer of this work is M. Y. Name.
\% This work consists of the files pig.dtx and pig.ins
\% and the derived file pig.sty.
\end{verbatim}`

Given such a notice and statement in a file, the conditions given in this license document would apply, with the ‘Work’ referring to the three files ‘pig.dtx’, ‘pig.ins’, and ‘pig.sty’ (the last being generated from ‘pig.dtx’ using ‘pig.ins’), the ‘Base Interpreter’ referring to any \TeX{}-Format, and both ‘Copyright Holder’ and ‘Current Maintainer’ referring to the person ‘M. Y. Name’.

If you do not want the Maintenance section of \LPPL{} to apply to your Work, change ‘maintained’ above into ‘author-maintained’. However, we recommend that you use ‘maintained’ as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

### Derived Works That Are Not Replacements

Several clauses of the \LPPL{} specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case
(e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

**Important Recommendations**

**Defining What Constitutes the Work**

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.