The \texttt{microtype} package

Subliminal refinements towards typographical perfection

R Schlicht
\texttt{w.m.l@gmx.net}

v2.8
2020/12/07

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

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

The alternative package \texttt{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 © 2004–2020 R Schlicht. It may be distributed and/or modified under the conditions of the \texttt{EPoX 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 9
   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
      Compatibility [51]
   14.2 Font setup 56
      Tracking [74] Disabling ligatures [85] Loading the configuration [87]
      Translating characters into slots [91] Hook into \LaTeX{}'s font selection [99]
      Context-sensitive setup [102]
   14.3 Configuration 104
      Fine tuning [112] Character inheritance [119] Permutation [121]
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</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>89</td>
</tr>
</tbody>
</table>

---

14.4 Package options ............................... 124

15 Configuration files 142
15.1 Font sets ........................................ 142
15.2 Font variants and aliases .......................... 143
15.3 Interaction with babel ............................ 145
15.4 Note on admissible characters ...................... 145
15.5 Character inheritance ............................ 146
  Euro symbols [150]
15.6 Tracking .......................................... 150
15.7 Font expansion ...................................... 151
15.8 Character protrusion ................................ 153
  Euro symbols [193]
15.9 Interword spacing .................................. 194
  Nonfrenchspacing [196]
15.10 Additional kerning ................................ 197
  French [197] Turkish [198]

16 OpenType configuration files 199
16.1 Character inheritance ................................ 199
16.2 Character protrusion ................................ 203

17 Auxiliary file for micro fine tuning 216
A The title logo ....................................... 218
A.1 Macros .............................................. 218
A.2 Document ........................................... 223
B The letterspacing illustration 223
B.1 Macros .............................................. 223
B.2 Document ........................................... 226
C Change history ....................................... 228
D Index ................................................ 237
E The LaTeX Project Public License 246

---
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 \pdf\TeX, and have since also propagated to \Lu\TeX and \Xe\TeX. 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 Hán Thế Thành, the author of \pdf\TeX, 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.\footnote{Thanh 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 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 exten-
sions. All micro-typographic aspects may be customised to your taste and needs in
a straight-forward and systematic manner. The next chapters present a survey of
all options and customisation possibilities. Should the micro-typographic extension
discussed in a section work only with certain \TeX engines, this requirement is
marked inside a grey text box on the right.

2 Getting started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

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

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

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

3.1 Enabling the micro-typographic features

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>protrusion</td>
<td>true, false, compatibility, nocompatibility, {font set name}</td>
<td>true</td>
</tr>
<tr>
<td>expansion</td>
<td>true, false, {font set name}</td>
<td></td>
</tr>
<tr>
<td>activate</td>
<td>true, false, {font set name}</td>
<td></td>
</tr>
</tbody>
</table>

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

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

\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 compatibility to the protrusion and/or expansion options. Typographically, however, the results will be suboptimal, hence the default value is nocompatibility.

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

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>tracking</td>
<td>true, false, {font set name}</td>
<td>false</td>
</tr>
<tr>
<td>kerning</td>
<td>true, false, {font set name}</td>
<td>false</td>
</tr>
<tr>
<td>spacing</td>
<td>true, false, {font set name}</td>
<td></td>
</tr>
</tbody>
</table>

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

These features do not unconditionally improve the quality of the typeset text: the spacing feature is still considered experimental, while the kerning feature only makes sense in special cases. Therefore, neither feature is enabled by default. They are not available with \XeTeX{} or \LuaTeX{}. 
Table 1: Availability of micro-typographic features

<table>
<thead>
<tr>
<th>Engine</th>
<th>Version</th>
<th>Output</th>
<th>Protrusion</th>
<th>Expansion</th>
<th>(≠ auto)</th>
<th>Kerning</th>
<th>Spacing</th>
<th>Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdfTeX</td>
<td>&lt; 0.14f</td>
<td>DVI/PDF</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td></td>
<td>≥ 0.14f</td>
<td>DVI/PDF</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td></td>
<td>≥ 1.20</td>
<td>DVI</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDF</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td></td>
<td>≥ 1.40</td>
<td>DVI</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDF</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
</tr>
<tr>
<td>LuaTeX</td>
<td>≥ 0.30</td>
<td>DVI</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>ø</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDF</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>ø</td>
</tr>
<tr>
<td></td>
<td>≥ 0.62</td>
<td>DVI</td>
<td>★</td>
<td>Ø</td>
<td>(ø)²</td>
<td>Ø</td>
<td>Ø</td>
<td>ø</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PDF</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>ø</td>
</tr>
<tr>
<td>XeTeX</td>
<td>≥ 0.9997</td>
<td>PDF</td>
<td>★</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>Ø</td>
<td>ø</td>
</tr>
</tbody>
</table>

★ = enabled   Ø = not enabled   ø = not available   a = by means of variable tracking

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

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

### 3.2 Character protrusion

**factor (integer)**

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

**unit character, (dimension)**

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

### 3.3 Font expansion

**auto true, false**

Beginning with pdfTeX version 1.20 (inherited by LuaTeX), the expanded instances of the fonts may be calculated automatically and at run-time instead of the user having to prepare them in advance. This option is true by default provided that you
are using a \TeX engine with this capability and the output mode is PDF.\(^2\) If auto is set to false, the font instances for all expansion steps must exist (with files called \texttt{〈font name〉+〈expansion value〉}, e.g., \texttt{cmr12+10}, as described in the pdf\TeX manual). With Lua\TeX, expansion is always automatic.

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

\begin{verbatim}
stretch 〈integer〉 20
shrink  You may specify the stretchability and shrinkability of a font, i.e., the maximum amount that a font may be stretched or shrunk. The numbers will be divided by 1000, so that a stretch limit of 10 means that the font may be expanded by up to 1\%. The default stretch limit is 20. The shrink limit will by default be the same as the stretch limit.
step 〈integer〉 * 1
Fonts are not expanded by arbitrary amounts but only by certain discrete steps within the expansion limits. With recent versions of pdf\TeX (1.40 or newer) or Lua\TeX, this option is by default set to 1, in order to allow trying the maximum number of font instances, and hence to guarantee the best possible output.\(^3\) Older pdf\TeX 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 pdf\TeX version, step is by default set to one fifth of the smaller value of stretch and shrink.
selected true, false false
When applying font expansion, it is possible to restrict the expansion of some characters that are more sensitive to deformation than others (e.g., the 'O', in contrast to the '|''). This is called \textit{selected expansion}, and its usage allows increasing the stretch and shrink limits (to, say, 30 instead of 20); however, the gain is limited since at the same time the average stretch variance will be decreased. Therefore, this option is by default set to \texttt{false}, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.
\end{verbatim}

\subsection*{3.4 Tracking}

\begin{verbatim}
letterspace 〈integer〉 100
This option changes the default amount for tracking (see section 5.3) resp. letter-spacing (see section 7). The amount is specified in thousandths of 1em; admissible values are in the range of \(-1000\) to \(+1000\).
\end{verbatim}

\(^2\) With pdf\TeX, \textit{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 cm-super package or use the Latin Modern fonts (package lmodern).

\(^3\) The downside with this default is that pdf\TeX 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}.
3.5 Miscellaneous options

draft \text{true, false} \quad \text{false}

If the \texttt{draft} option is passed to the package, \textit{all micro-typographic extensions will be disabled}, which may lead to different line, and hence page, breaks. The \texttt{draft} and \texttt{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 \texttt{draft} to show any overfull boxes, you should load \texttt{microtype} with the \texttt{final} option.

\verb|verbose| \text{true, false, errors, silent} \quad \text{false}

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

\verb|babel| \text{true, false} \quad \text{false}

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

\verb|config| \text{(file name)} \quad \text{microtype}

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

\verb|DVIoutput| \text{true, false} \quad \text{false}

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

3.6 Changing options later

\texttt{\microtypesetup} \{(key = value list)\}

Inside the preamble, this command accepts all package options described above (except for \texttt{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: \texttt{expansion}, \texttt{protrusion} and \texttt{activate}, which in turn may receive the values \texttt{true, false, compatibility or nocompatibility}, and \texttt{tracking, kerning} and \texttt{spacing} with the admissible values \texttt{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:

\texttt{\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.

\begin{verbatim}
\DeclareMicrotypeSet \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 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 \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,T51,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 \(\text{value}\)default, e.g., \rmdefault. A single asterisk means \(\text{attribute}\)default, e.g., \encodingdefault, respectively \normalsize for the size axis. Sizes may either be specified as a dimension (‘10’ or ‘10pt’), or as a size selection command without the backslash. You may also specify ranges (e.g., ‘\small-Large’); while the lower

\footnote{These translations will take place at\begin{document}, which means that changes to the defaults inside the preamble will also be taken into account. Only in cases where you change font defaults at\begin{document} yourself, you need to load 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</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 (\normalsize)</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}
\UseMicrotypeSet \{ \{features\} \} \{ \{set name\} \}

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

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

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

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

5 Micro fine tuning

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

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

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

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

\SetProtrusion \[ \text{[options]} \] \{ \text{set of fonts} \} \{ \text{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 \{ \text{encoding} = \text{T1}, \text{family} = \text{cmr} \} \{ \text{A} = \{50,50\}, \text{\textquoteleft} = \{700,\} \} \]

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

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

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

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

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

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

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

5.2 Font expansion

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

Options:

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

\(^5\)The unit option can even be passed globally to the package (cf. section 3.2). However, all provided settings are created under the assumption that the values are relative to the character width. Therefore, you should only change it if you are certain that the default settings will not be used in your document.
all expansion factors, to set the input encoding, or to determine the context of the list (expansion contexts are only possible with pdfTeX version 1.40.4 or newer).

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

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

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

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

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

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

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

## Tracking

An important typographic technique – which was missing in Ti\TeX{} for a long time – is the adjustment of tracking, i.e., the uniform addition or subtraction of letter space. For older versions, a dirty trick is laid out in section 14.2 on page 58.
to/from all the characters in a font. For example, it is good typographic practice
to slightly space out text set in all capitals or small capitals (as in this document).
Legibility may also be improved by minimally increasing the tracking of smaller
and decreasing that of larger type.\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 $\texttt{\textbackslash SetTracking}$ command allows specifying
the tracking amount for different fonts or font sets. It will also be evaluated by the
$\texttt{\textbackslash textls}$ command, which may be used for letterspacing shorter pieces of text (see
section 7).

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

\textbf{Options:}

\begin{itemize}
\item \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.
\item \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}
\texttt{\textbackslash SetTracking[ \texttt{spacing}=\{25*,166,}\}\}\{ \texttt{encoding}=*, \texttt{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.
\item \texttt{outer spacing} If an interword space immediately precedes or follows
letterspaced text, it will by default be equal to that within the text. With this option,
which accepts the same values as \texttt{spacing}, it may be adjusted independently.
\item \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 ‘\texttt{outer kerning}={0,0}’.
\item \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, \textit{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’,
\end{itemize}
‘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:

Stop stealing sheep!

While the word ‘Stop’ is not letterspaced, the space between the letters in the other two words is expanded by the tracking amount of 160/1000em = 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:

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

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

---

8 With pd\TeX versions older than 1.40.4, all ligatures, and hence all kerning, will be disabled. It is therefore recommended to use at least version 1.40.4.
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 differently, this feature allows modifying the left or right sidebearings of specific glyphs.)

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

The kerning settings are specified as pairs of \langle character \rangle = \langle 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 (dimension). 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
\[ name \text{ = french-default,} \]
\[ context \text{ = french,} \]
\[ unit \text{ = space } \]
\[ \{ encoding \text{ = \{OT1,T1,LY1\} } \]
\[ \{ \text{ : = \{1000,\}, }\% = \text{ \fontdimen2}\]
\[ \; = \{500,\}, }\% - \text{ \thinspace}\]
\[ ! = \{500,\}, }\]
\[ ? = \{500,\} }\]

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

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

Options:

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

unit You can specify the unit by which the specified numbers are measured. Possible values are: character, a (dimension) and, additionally, space. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with the following (nonsensical) settings:
the space inserted after a full stop would be doubled (technically speaking: \(2 \times \fontdimen{}\)) as would the maximum stretch and shrink amounts of the interword space (\(\fontdimen{}\) and 4). Conversely, setting all three values to −1000 would completely cancel a space after the respective character.

5.6 Character inheritance

\[\text{\textbackslash DeclareCharacterInheritance}\{\{\text{features}\}\}\{\{\text{set of fonts}\}\}\{\{\text{inheritance lists}\}\}\]

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters À, Á, Â, Ã, Ä, Å and ˘A should probably be protruded by the same (absolute) amount as the character A. Using the command \texttt{\textbackslash 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 \texttt{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 \{\texttt{base character}\} = \{\texttt{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 \LaTeX{} and \TeX{}E, 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).

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 ‘\texttt{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: ‘\texttt{mt-\{}font family\}\texttt{.cfg}’ (e.g., ‘\texttt{mt-cmr.cfg}’); any spaces in the font name should be removed, e.g., ‘\texttt{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.

\[\text{\textbackslash DeclareMicrotypeVariants}\{\{\text{list of suffixes}\}\}\]

\[\text{\textbackslash DeclareMicrotypeVariants*}\]

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 fonts \texttt{padx} (expert set), \texttt{padj} (oldstyle numerals) and \texttt{pad} (plain) into one and the
Table 3: Font settings

<table>
<thead>
<tr>
<th>Font family (NFSS code)</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encodings [Scripts]</td>
</tr>
</tbody>
</table>
| Generic                 | OT1, T1, T2A, LY1, QX, (TS1)
| Computer Modern Roman (cmr) | OT1, OT4, T1, T2A, T5, LY1, TS1 |
| Bitstream Charter (bch)  | OT1, T1, T5, LY1, TS1 |
| Adobe Garamond (pad, padx, padj) | OT1, T1, LY1, TS1 |
| URW Garamond (ugm)      | OT1, T1, TS1 |
| Bitstream Letter Gothic (blg) | OT1, T1, TS1 |
| Adobe Minion (pmmx, pmmj) | OT1, T1, T2A, LY1, TS1 |
| Palatino (plp, plpx, plplj) | OT1, OT4, T1, LY1, (TS1) |
| Times (ptm, ptmx, ptmj)  | OT1, OT4, T1, LY1, QX, (TS1) |
| Latin Modern Roman      | EU1/2, TU [Latin, Greek] |
| Charis SIL              | EU1/2, TU [Latin, Cyrillic, Greek] |
| Palatino Linotype       | EU1/2, TU [Latin] |
| Computer Modern math (cmsy, cmm) | OML/OMS |
| AMS symbols (msa, msb)   | U |
| Euler (eur, eus, euf)    | U |
| Euro symbols (Adobe, ITC, marvosym) | U/OT1 |

a Incomplete
b Aliases: Latin Modern Roman (1mr), ae (aer), zefonts (zerr), eco (cmor), hfoldsty (hfor)
c Aliases: mathdesign/Charter (mdch), MicroPress's cmath (chr), Xcharter
d Settings inherited from italic shape
e Aliases: mathdesign/URW Garamond (ndugm), garamondx (zgmx, zgjm)
f Alias: ulgothic (ulg)
g Aliases: pxfonts (pav), qfonts/QuasiPalatino, T\TeX~Gyre~Pagella (qp1), newpx, FPL~Neu (fp9x, fp9j),
domitian
h Aliases: txfonts (txr), qfonts/QuasiTimes, T\TeX~Gyre~Termes (qtm), newtxt, tempora, step, stix/stix2
i Alias: New Computer Modern
j Aliases: T\TeX~Gyre~Pagella, Palatino LT Std, Palatino, Domitian
k Aliases: Latin Modern (lns, lnm)
l Alias: eulerm (zeur, zeus)

same file mt-pad.cfg. This command expects a comma-separated list of variant suffixes. The starred version appends the suffix(es) to the existing list. The default declaration in microtype.cfg is:

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

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{1mr}{cmr}
```

which would make the package, whenever it encounters the font 1mr 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 \{\textit{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{\textbackslash Microtype\textbackslash Hook}, described in the implementation part, section 14.4.4.} This command will load the file ‘mt-\{\textit{font name}\}.cfg’.

6 \textbf{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 \{\textit{context assignments}\}

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

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

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

\textmicrotypecontext \{\textit{general text}\}

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

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

\SetProtrusion
[\texttt{context} = footnote ]
\{\texttt{font} = \{.650\}, \texttt{2} = \{.400\}, \texttt{3} = \{.400\}, \texttt{4} = \{.400\}, \texttt{5} = \{.400\}, \texttt{6} = \{.400\}, \texttt{7} = \{.500\}, \texttt{8} = \{.400\}, \texttt{9} = \{.400\}, \texttt{0} = \{.400\}\}

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

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

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

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

\microtypecontext{kerning=french}

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

\textmicrotypecontext{kerning=}{English text!}

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

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

\[\text{\DeclarerMicrotypeBabelHook}\{\text{list of babel languages}\}\{\text{context list}\}\]

Naturally, \texttt{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:

\texttt{\DeclarerMicrotypeBabelHook}\{\texttt{french,francais,acadian,canadien}\}\{\texttt{kerning=french,}\}\texttt{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 \texttt{babel} option. Currently, \texttt{microtype} supports French and Turkish kerning and English spacing (aka. \texttt{nonfrenchspacing}). For unknown languages, all contexts will be reset.

\section{Letterspacing revisited}

While the \texttt{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, \texttt{microtype} introduces two commands that can be used (independently of whether the \texttt{tracking} option is enabled) in the same way as \LaTeX's text commands: \texttt{\textls{amount}} – 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.
Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you need not worry about protecting or breaking ligatures with most fonts. However, in certain situations, there may be a conflict of ligatures beginning with the same letter, where some of them should be inhibited, while others should not. When letterspacing text typeset in Fraktur fonts, for example, the ligatures ‘ch’, ‘ck’, ‘tz’ and ‘sz’ (‘ß’) should never be broken up; you also usually see the ‘st’ (‘ßt’) ligature in letterspaced text. Furthermore, at least the `yfonts` package realises the short s (‘s’) as the ligature ‘ss’. 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, ‘\$s\textls{l|t}s\textls{i|g}k\textls{e}it’, with ligatures shown in green, inhibited ligatures in red).

\begin{verbatim}
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{120}
\textfrak{Aus:s\textls{icht}s\textls{l|os}\textls{igkeit}}
\end{verbatim}

\begin{verbatim}
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{120}
\textfrak{Au\lslig{s}si\lslig{ch}s\lslig{los}igkeit}
\end{verbatim}

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

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

\begin{verbatim}
\DisableLigatures{encoding = T1, family = tt*}
\end{verbatim}
It is also possible to disable selected ligatures only. The optional argument may contain a comma-separated list of characters for which the ligature mechanism should be inhibited:

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

Only the character that begins the ligature(s) should be specified. This command may only be used in the preamble, and only once.\footnote{With Lua\TeX{}, 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 Lua\TeX{}, 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 \texttt{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 \texttt{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 \texttt{\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 \texttt{verbatim} environment frequently. The following line (which requires the etoolbox package), added to the document’s preamble, would serve the same purpose:

\BeginEnvironment{verbatim}{\microtypesetup{activate=false}}

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

\textit{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.

\textit{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.

\textit{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 \texttt{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.

\textit{Compatibility and interaction with other packages:} The \texttt{microtype} package is supposed to work happily together with all other \LaTeX{} packages (except for pdfcrop). 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 \texttt{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 \texttt{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 fonts pkg 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.\footnote{12 They make use of features provided by luaotfload (via fonts pkg).}

• 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 or LuaTeX in PDF mode. Although expansion is also possible in DVI mode with pdfTeX, it requires that all instances of the expanded fonts exist on your TeX system.

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

* Warning: pdflatex: font ptmrm8r cannot be expanded (not an included Type1 font)
and the PDF viewer complains about a missing font, e.g., Adobe Reader thusly:

\texttt{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.
ACKNOWLEDGMENTS

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

Furthermore, pdfTeX versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdfTeX versions, this is only possible if you manually create expanded instances of the fonts.

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

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

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

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

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

This warning will occur with pdfTeX versions older than 1.40.4, if tracking and expansion is applied to a font. It is harmless and can be ignored.

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

10 Contributions

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

11 Acknowledgments

This package would be pointless if H\’an Th\’anh 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 Th\’anh 2004, Th\’anh 2008 and elsewhere. I also thank him and the rest of the pdfTeX team, and more recently also the Lua\TeX and X\TeX teams, for refuting the idea that \TeX is dead, and for fixing the bugs I find.
Harald Harders has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. Andreas Bühmann has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. Ulrich Dirr has made numerous suggestion, especially concerning the new extensions of interword spacing adjustment and additional character kerning. Georg Duffner has patiently tested microtype under Xe\TeX\ and Lua\TeX\ with his beautiful OpenType font EB Garamond\(^\text{13}\). My thanks also go to Maciej Eder for contributing settings for the QX encoding, as well as to Karl Karlsson for providing settings for the Cyrillic T2A encoding, and to Hendrik Vogt, who made substantial improvements to the Computer Modern Roman italic settings. I thank Loren B. Davis for providing protrusion settings for OpenType versions of Palatino Linotype. I am also very much indebted to Élie Roux, who not only contributed the 1ua 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 csquotes package the possibility to restore the original meanings of all activated characters, thus allowing for these characters to be used in the configuration files. Peter Wilson kindly provided a hook in his ledmac/ledpar packages, so that critical editions can finally also benefit from character protrusion. Likewise, Donald Arseneau patched his shapepar package to accommodate protrusion.


**References**


\(^\text{13}\) Available from CTAN at \url{pkg/ebgaramond}, including configuration files for microtype.


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

13 Short history

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

2.8 (2020/12/07)
• New default font sets for expansion and spacing: ‘alltext-nott’ [4, table 2]

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

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

2.5 (2013/03/13)
• Support for the fontspec package, viz. for OpenType fonts with LuaTeX and XeLaTeX
• Support for protrusion with XeLaTeX ≥ 0.9997
• Support for tracking/letterspacing with Lua\TeX\ 0.62
• Allow context-sensitive setup with Lua\TeX\ 
• 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 pdf\TeX\ 1.40 [3.3]

2.3c (2008/11/11)
• Support for Lua\TeX\ 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 (pdf\TeX\ 1.40.4); automatically adjust protrusion settings
• New key 'no ligatures' for \SetTracking to disable selected or all ligatures (pdf\TeX\ 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 (pdf\TeX\ 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 pdf\TeX\ 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 TS encoded Charter

1.9 (2005/10/28)
• New command \DisableLigatures to disable ligatures (pdf\TeX ≥ 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 ≥ 1.30)

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

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

1.5 (2004/12/15)
• When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled \[3.1\]
• New option 'selected' to enable selected expansion, default: false \[3.3, 5.2\]
• New default for expansion option 'step': 4 \((\min(\text{stretch, shrink})/5)\) \[3.3\]
• 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' \[4, \text{table 2}\]
• 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: \texttt{\textbackslash DeclareCharacterInheritance} \[5.6\]
• Characters may also be specified as octal or hexadecimal numbers \[5\]

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

The docstrip modules in this file are:

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

package: The code for the microtype package (microtype.sty).
pdftex-def: Definitions specific to pdfTeX (microtype-pdftex.def).
xetex-def: Definitions specific to XeTeX (microtype-xetex.def).
luatex-def: Definitions specific to LuaTeX (microtype-luatex.def).

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

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

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

luafile: Lua functions (microtype.lua).

config: Surrounds all configuration modules.

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

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

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

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

And now for something completely different.
14.1 Preliminaries

This is us.

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.

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

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 pdflatex.
These commands also have a starred version.

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.

Don't load letterspace.

The old command names had one more hunch.

Cases for `\tracingmicrotype`:

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

14.1.1 Debugging

Communicate.

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.)

`\newcommand*{\islig}[1]{{}}`

These commands also have a starred version.

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.

Don't load letterspace.

The old command names had one more hunch.

Cases for `\tracingmicrotype`:

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

14.1.1 Debugging

Communicate.

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.)

`\newcommand*{\islig}[1]{{}}`
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 \texttt{tracingmicrotypeinpdf}:

1: show new fonts
2: + show known fonts

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

During font setup, we save the text for the popup in \texttt{MT\@pdf@annot}. (This requires \texttt{pdf\TeX} $\geq 1.30$.) The \texttt{pdftexcmds} package provides \texttt{pdf\TeX}’s utility commands in \texttt{Lua\TeX}, too.

With \texttt{tracingmicrotypeinpdffalse}, 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 \texttt{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).
14.1.2 Requirements

The `letterspace` package works with:

0: miniltx
1: eplain
2: LaTeX

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

For definitions that depend on e-TEX features.

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

If we are using neither of these engines, we disable everything and exit.
A hack circumventing the \TeX{} Live 2004 hack which undefines the pdf\TeX{} primitives in the format in order to hide the fact that pdf\TeX{} is being run from the user. This has been fixed in \TeX{} Live 2005.

\begin{verbatim}
\ifx\normalpdftexversion\@undefined \else
  \let\pdftexversion \normalpdftexversion
  \let\pdftexrevision \normalpdftexrevision
  \let\pdfoutput \normalpdfoutput
\fi
\end{verbatim}

Old packages might have let \texttt{\pdftexversion} to \texttt{\relax}.

\begin{verbatim}
\let\MT@engine\relax
\MT@engine@tooold
\ifx\pdftexversion\@undefined \else
  \ifx\pdftexversion\relax \else
    \def\MT@engine{pdf}
    \def\MT@pdf@or@lua\@firstoftwo
    \ifnum\pdftexversion > 139 \def\MT@engine@tooold{1}\fi
  \fi
\fi
\ifx\directlua\@undefined \else
  \ifx\directlua\relax \else
    \def\MT@engine{lua}
    \ifnum\luatexversion < 62 \def\MT@engine@tooold{0}\else
      \def\MT@engine@tooold{1}
      \let\MT@lua\directlua
      \ifnum\luatexversion > 84
        \let\pdfoutput\outputmode
        \let\pdfprotrudechars\protrudechars
        \let\pdfadjustspacing\adjustspacing
      \fi
    \fi
  \fi
\fi
\ifx\XeTeXversion\@undefined \else
  \ifx\XeTeXversion\relax \else
    \def\MT@engine{xe}
  \fi
\fi
\end{verbatim}

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. \texttt{\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 $\mathrm{efcode} = 1000$ ($\geq 1.20$)

5: + $(\text{left, right})\text{marginkern}; \text{pdfnoligatures}; \text{pdfstrcmp}; \text{pdfescapestring}$ ($\geq 1.30$)

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

7: + \text{letterspacefont} doesn’t disable ligatures and kerns; \text{pdfcopyfont}$ ($\geq 1.40.4$)

---

$\MT@xetex@no$

\LaTeXeX supports character protrusion since version 0.9997.

$\MT@luatex@no$

Cases for Lua\TeX ($\\luatexversion$ ought to have been enabled by the format):

0: N/A

1: Lua\TeX ($< 0.36$)

2: + \text{\directlua} without state number ($\geq 0.36$)

---

14 This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.
3: \letterspacefont; non-automatic expansion doesn’t work anymore, and automatic expansion in DVI mode is realised by modifying the tracking, not the glyphs (≥ 0.62)

4: almost all of the pdf\TeX primitives have been renamed (≥ 0.85)

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

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

\Communicate with lua. Beginning with Lua\TeX 0.36, \directlua no longer requires a state number.

\let\MT@lua\directlua
\def\MT@luatex@no{5}
\ifnum\luatexversion<90
\def\MT@luatex@no{4}
\ifnum\luatexversion<85
\def\MT@luatex@no{3}
\ifnum\luatexversion<62
\def\MT@luatex@no{2}
\ifnum\luatexversion<36
\def\MT@lua{0}
\def\MT@luatex@no{1}
\fi
\fi
\fi
\fi
\MT@dinfo@nl{0}{luatex no.: \MT@luatex@no}
\MT@clear@options
\endinput

Still there? Then we can begin: We need the keyval package, including the ‘new’ \KV@@sp@def implementation.
\RequirePackage{keyval}[1997/11/10]
\newtoks\MT@toks
\ifMT@if@
\newif\ifMT@if@
\MT@toks
\MT@if@protrusion\MT@if@expansion\MT@if@auto\MT@if@selected
\MT@if@noligatures\MT@if@draft\MT@if@spacing\MT@if@kerning
\MT@if@tracking\MT@if@babel
\MT@protrusion\MT@expansion\MT@auto\MT@selected
\MT@noligatures\MT@draft\MT@spacing\MT@kerning\MT@tracking
\MT@babel
[This line intentionally left blank.]
\MT@pr@level\MT@ex@level\MT@pr@factor\MT@ex@factor
\MT@pr@unit\MT@sp@unit\MT@kn@unit\MT@shrink\MT@step
\MT@pr@min\MT@pr@max\MT@ex@min\MT@ex@max\MT@sp@min\MT@sp@max
\MT@kn@min\MT@kn@max\MT@tr@min\MT@tr@max
\MT@factor@default\MT@stretch@default\MT@shrink@default
\MT@pr@min\MT@pr@max\MT@ex@min\MT@ex@max\MT@sp@min\MT@sp@max
\MT@kn@min\MT@kn@max\MT@tr@min\MT@tr@max
\MT@factor@default\MT@stretch@default\MT@shrink@default

\MT@toks

We need a token register.
\newtoks\MT@toks
\ifMT@if@
\newif\ifMT@if@
\MT@toks
\MT@if@protrusion\MT@if@expansion\MT@if@auto\MT@if@selected
\MT@if@noligatures\MT@if@draft\MT@if@spacing\MT@if@kerning
\MT@if@tracking\MT@if@babel
\MT@protrusion\MT@expansion\MT@auto\MT@selected
\MT@noligatures\MT@draft\MT@spacing\MT@kerning\MT@tracking
\MT@babel
[This line intentionally left blank.]
\MT@pr@level\MT@ex@level\MT@pr@factor\MT@ex@factor
\MT@pr@unit\MT@sp@unit\MT@kn@unit\MT@shrink\MT@step
\MT@pr@min\MT@pr@max\MT@ex@min\MT@ex@max\MT@sp@min\MT@sp@max
\MT@kn@min\MT@kn@max\MT@tr@min\MT@tr@max
\MT@factor@default\MT@stretch@default\MT@shrink@default
\MT@pr@min\MT@pr@max\MT@ex@min\MT@ex@max\MT@sp@min\MT@sp@max
\MT@kn@min\MT@kn@max\MT@tr@min\MT@tr@max
\MT@factor@default\MT@stretch@default\MT@shrink@default

\MT@toks

A scratch if.
\MT@letterspace \def\MT@letterspace@default{20}

\MT@letterspace@default Default value for letterspacing (in thousandths of 1em).

\if\MT@document Our private test whether we're still in the preamble.

14.1.4 Auxiliary macros

\MT@requires@pdftex \MT@requires@luatex For definitions that depend on a particular pdf\TeX\ resp. Lua\TeX\ version.

\selectlanguage{english}
Some functions are loaded from a dedicated lua file. This avoids character escaping problems and incompatibilities between versions of Lua\TeX. Unless running a recent \TeX, we load the luatexbase package.

\if\fmtversion{2016/01/01} \relax \RequirePackage{luatexbase}\fi

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 Lua\TeX\ documents that don't load this package, anyway. Since 2017/01/01, it is already loaded in the format.

\if\fmtversion{2017/01/01} \relax \RequirePackage{luaotfload}\fi

\MT@lua{require("microtype")}

Here it begins. The module was contributed by Élie Roux.
To be continued, but first back to primitives.

Here's the forgotten one.

Commands to create command sequences. Those that are going to be defined globally should be created inside a group so that the save stack won't explode.

This is \texttt{\verb|\def|} and global.

Its expanding versions.

\texttt{\verb|\def|} a \texttt{|sequence} to a \texttt{|command|}.

\texttt{\verb|\def|} a \texttt{|command|} to a \texttt{|sequence|}.

Expand the second token once and enclose it in braces.

Expand the next two tokens after \texttt{\{#1\}} once.

Expand the next two tokens after \texttt{\{#1\}} once and enclose them in braces.

You do not wonder why \texttt{\verb|\def|} doesn't exist, do you?

Wrapper for testing whether command resp. \texttt{|sequence|} is defined. If we are running \LaTeX, we will use its primitives \texttt{\ifdefined} and \texttt{\ifcsname}, which decreases memory use substantially.

You do not wonder why \texttt{\verb|\def|} doesn't exist, do you?

Wrapper for testing whether command resp. \texttt{|sequence|} is defined. If we are running \LaTeX, we will use its primitives \texttt{\ifdefined} and \texttt{\ifcsname}, which decreases memory use substantially.

You do not wonder why \texttt{\verb|\def|} doesn't exist, do you?
\MT@detokenize@n Translate a macro into a token list. With e-T\TeX, we can use \detokenize. We also need to remove the last trailing space; and only the last one – therefore the fiddling (and the \string isn't perfect, of course).

\MT@ifempty Test whether argument is empty.

\MT@ifint Test whether argument is an integer, using an old trick by Mr. Arseneau, or the latest and greatest from pdfT\TeX or LuaT\TeX (which also allows negative numbers, as required by the \texttt{letterspace} option).
\MT@ifint#1{"\ifnum9<1#1!\else?\fi\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi"}
\MT@ifdimen
Test whether argument is dimension (or number). (nd and nc are new Didot resp. Cicero, added in pdiTex 1.30; px is a pixel.)
\MT@ifdim \ Compare floating point numbers.
\MT@ifstreq \ Test whether two strings (fully expanded) are equal.
\MT@xadd \ Add item to a list.
\MT@xaddb \ Add item to the beginning.
Run \( \#2 \) on all elements of the comma list \( \#1 \). This and the following is modelled after \LaTeX{} commands.

\begin{verbatim}
\def\MT@map@clist@n#1#2{\ifx\@empty#1\else\def\MT@clist@function##1{#2}\MT@map@clist@#1,\@nil,\@nnil}\fi}
\def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
\def\MT@map@clist@#1,{\ifx\@nil#1\expandafter\MT@clist@break\fi\MT@clist@function{#1}\MT@map@clist@}
\let\MT@clist@function\@gobble
\def\MT@clist@break#1\@nnil{}\end{verbatim}

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

\begin{verbatim}
\def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
\def\MT@map@tlist@#1#2{\ifx\@nnil#2\else#1{#2}\expandafter\MT@map@tlist@\expandafter#1\fi}
\def\MT@tlist@break#1\@nnil{\fi}
\end{verbatim}

Test whether item \( \#1 \) is in comma list \( \#2 \). Using \texttt{pdfmatch} would be slower.

\begin{verbatim}
\newif\MT@in@clist
\def\MT@in@clist#1#2{\def\MT@res@a##1,#1,##2##3\@nnil{\ifx##2\@empty\MT@inlist@false\else\MT@inlist@true\fi}\expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil}
\MT@rem@from@clist
\def\MT@rem@from@clist#1#2{\def\MT@res@a##1,#1,##2\MT@res@a{##1,##2\MT@res@b}\def\MT@res@b##1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}\xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}}
\MT@in@tlist
\MT@in@tlist@
\end{verbatim}

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

```latex
\def\MT@in@tlist#1#2{\MT@inlist@false\def\MT@res@a{#1}\MT@map@tlist@c#2\MT@in@tlist@}
\def\MT@in@tlist@#1{\edef\MT@res@b{#1}\ifx\MT@res@a\MT@res@b\MT@inlist@true\expandafter\MT@tlist@break\fi}
\MT@in@rlist\MT@in@rlist@\MT@in@rlist@@\MT@size@name
\def\MT@in@rlist#1{\MT@inlist@false\MT@map@tlist@c#1\MT@in@rlist@}
\def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
\def\MT@in@rlist@@#1#2#3{\MT@ifdim{#2}=\m@ne{\MT@ifdim{#1}=\MT@size\MT@inlist@true\relax}{}{\MT@ifdim\MT@size<#1\relax{}{\MT@ifdim\MT@size<#2\MT@inlist@true\relax{}{\ifMT@inlist@\def\MT@size@name{#3}\expandafter\MT@tlist@break}\fi}}}
\MT@loop\MT@iterate\MT@repeat
\def\MT@loop#1\MT@repeat{\def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}}\MT@iterate\let\MT@iterate\relax\let\MT@repeat\fi
\MT@while@num
\def\MT@while@num#1#2#3{\@tempcnta#1\relax\MT@loop #3\advance\@tempcnta \@ne\ifnum\@tempcnta < #2\MT@repeat}}
\MT@if@luaotf@font
\MT@pdf@or@lua{\let\MT@if@luaotf@font\@secondoftwo}{\def\MT@if@luaotf@font{\csname\MT@lua{\luatex-def}{microtype.if_luaotf_font()}}}
\local function if_luaotf_font()
local thefont = font.getfont(font.current())

\MT@loop\MT@iterate\MT@repeat
\MT@while@num
\MT@if@luaotf@font
```

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

\texttt{\MT@loop} \texttt{\MT@iterate} \texttt{\MT@repeat}

Execute \texttt{(#3)} from \texttt{(#1)} up to (excluding) \texttt{(#2)} (much faster than \LaTeX's \texttt{\@whilenum}).

\texttt{\MT@while@num}

\texttt{\MT@if@luaotf@font}

For fonts loaded by \texttt{\luaotfload} we query the font's table.

\texttt{\MT@loop} \texttt{\MT@iterate} \texttt{\MT@repeat}}
\input{tex/impl_base}

\begin{verbatim}
\MT@do@font
Execute (#1) 256 times,
\end{verbatim}

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

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

\begin{verbatim}
\MT@count
Increment macro (#1) by one. Saves using up too many counters. The e-\TeX way is slightly faster.
\end{verbatim}

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

\texttt{\MT@do@font} resp. for the whole font for Lua\LaTeX, if it's a Unicode font.

\begin{verbatim}
\def\MT@do@font#1{%\MT@while@num\z@\@cclvi
\MT@if@luaotf@font{%\def\MT@dofont@function{#1}%\MT@lua{microtype.do_font()}%}{\MT@while@num\z@\@cclvi{#1}}%}
\end{verbatim}
Multiply and divide a counter. If we are using e-TeX, we will use its `\numexpr` primitive. This has the advantage that it is less likely to run into arithmetic overflow. The result of the division will be rounded instead of truncated. Therefore, we'll get a different (more accurate) result in about half of the cases.

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

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

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

These we also need the other way round.

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

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

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

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

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

### 14.1.5 Compatibility

For the record, the following e-TeX kernel commands will be modified by `microtype`:
The wordcount package redefines the font-switching commands, which will break microtype. Since microtype doesn't have an effect on the number of words in the document anyway, we will simply disable ourselves.

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

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

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

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

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

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

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

Two new conditionals for use with \TeX or Lua\TeX.

If \verb!*\tikz@expandcount* is greater than zero, we're inside or at the end of a \verb!tikz! node, where we don't want to adjust spacing after letterspacing, lest we disturb \verb!tikz!. This is used in \verb!\MT@afteraftergroup! and we don't need it for letterspace.
In the preamble, we check for the packages each time a font is set up. Thus, it will work regardless when the packages are loaded.

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

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

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

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

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

Check again at the end of the preamble.

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

The ‘pdfcprot’ package provides an interface to character protrusion. \{MT\} and ‘pdfcprot’ may not be used together!\MessageBreak
So does the \{MT\} package. Using both packages at the same time will almost certainly lead to undesired results. Have your choice!\%

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

\begin{verbatim}
\MT@iffalse
\MT@withpackage{T}{csquotes}{\MT@ifdefined@c{T}{\percentsign}{\let\%\percentsign}}\fi\MT@withpackage{T}{csquotes}{\@ifpackagelater{T}{csquotes}{2005/05/11}{\g@addto@macro\MT@setupfont@hook@disablequotes}{\MT@warning{Should you receive warnings about unknown slot\MessageBreak numbers, try upgrading the `csquotes' package}}}\fi\MT@iffalse
\MT@withpackage{T}{hyperref}{\pdfstringdefDisableCommands{\MT@ltx@pickupfont\let\textmicrotypecontext@secondoftwo\let\microtypecontext@gobble\MT@restore@p@h}}\MT@iftrue
\MT@withpackage{T}{tex4ht}\MT@iftrue
\MT@withpackage{T}{mathastext}\MT@iftrue
\MT@if\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
\end{verbatim}

The \texttt{listings} package makes numbers and letters active,
\begin{verbatim}
\MT@iffalse
\MT@withpackage{T}{listings}{\MT@ifdefined@c{T}{\space}{\let\space}\relax}\MT@iftrue
\MT@withpackage{T}{tex4ht}\MT@iftrue
\MT@withpackage{T}{mathastext}\MT@iftrue
\MT@if\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
\end{verbatim}

... and the backslash (which would lead to problems in \MT@get@slot).
\begin{verbatim}
\g@addto@macro\MT@cfg@catcodes{\MT@while@num{30}{3A}{\catcode@tempcnta=12}\MT@while@num{41}{5B}{\catcode@tempcnta=11}\MT@while@num{61}{7B}{\catcode@tempcnta=11}}\relax
\end{verbatim}

Inside a listing, \texttt{space} is redefined.
\begin{verbatim}
\def\space{ }\relax
\end{verbatim}

When loaded with the \texttt{extrudechar} option, \texttt{listings} will also redefine 8-bit active characters (\texttt{inputenc}). Luckily, this simple redefinition will make them expand to their original definition, so that they could be used in the configuration.
\begin{verbatim}
\MT@ltx@pickupfont\let\lst@ProcessLetter\@empty\relax
\end{verbatim}

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

\begin{verbatim}
\ifx\SOUL@undefined\else
  \soulregister\lsstyle 0\%
  \soulregister\textls 1\%
  \ifx\XeTeXrevision\undefined
    \let\MT@SOUL@doword\SOUL@doword
    \def\SOUL@doword{\pdfadjustspacing=\z@ \MT@SOUL@doword}%
  \fi
\fi
\end{verbatim}

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

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

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

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

\subsection*{Font setup}

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

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

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

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

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

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

\begin{verbatim}
\MT@setupfont@hook
\end{verbatim}

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

\begin{verbatim}
\MT@requires@pdftex{\MT@copy@font}
\end{verbatim}

The font properties must be extracted from \MT@font, since the current value of $\f@encoding$ and friends may be wrong!
Try to find a configuration file for the current font family.

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.

Interword spacing and kerning (\pdfTeX 1.40).

Disable ligatures (\pdfTeX 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.

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

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

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

We only need the font identifier for letterspacing.

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

Here's the promised dirty trick for users of older pdf\TeX\ 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
\[\begin{align*}
\text{stretch} &= 30, \\
\text{shrink} &= 60, \\
\text{step} &= 5 \\
\{ \text{encoding} = *, \\
\text{size} = 10.001 \} \\
\end{align*}\]

\newcommand{\expandpar}[1]{\% \\
\fontsize{10.001}{\baselineskip}\selectfont #1\par}\%
\expandpar{This paragraph contains an `unnecessary' widow.}

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

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

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

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

\MT@feat stores the current feature.

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

The generic test (\#1 is the axis, \#2 the feature, \@tempa contains the set name).

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

If no limitations have been specified, i.e., the list for a font attribute has not been defined at all, the font should be set up.

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

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

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

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.

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).

Get the name of the inheritance list and parse it.

Set an input encoding?
Load additional lists?
\MT@load@list\MT@pr@c@name
\MT@set@listname

Load the main list.
\MT@let@cn@tempc{MT@pr@c@MT@pr@c@name}\
\expandafter\MT@set@codes@tempc,\relax,\}
\MT@reset@pr@codes
\MT@set@all@pr
Set all protrusion codes of the font.
\MT@set@all@pr#1#2{\〈debug\MT@dinfo@nl{3}{-- lp/rp: setting all to #1/#2}\
\let\MT@temp\@empty\MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lpcode\MT@font\@tempcnta=#1}}\MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rpcode\MT@font\@tempcnta=#2}}\MT@do@font\MT@temp}
\MT@reset@pr@codes@\MT@reset@pr@codes
All protrusion codes are zero for new fonts. However, if we have to reload the font
due to different contexts, we have to reset them. This command will be changed by\microtypecontext if necessary.
\MT@reset@pr@codes0\MT@reset@pr@codes
\MT@the@pr@code\MT@the@pr@code@tr
If the font is letterspaced, we have to add half the letterspacing amount to the
margin kerns. This will be activated in \MT@set@tr@codes.
\MT@the@pr@code\MT@the@pr@code@tr
\MT@set@codes
Split up the values and set the codes.
\MT@set@codes#1,\{\%
\ifx\relax#1\empty\else\MT@split@codes #1==\relax\expandafter\MT@set@codes\fi
\MT@split@codes#1=#2=#3\relax{\def\@tempa{#1}\ifx\@tempa\@empty\else\MT@get@slot\ifnum\MT@char > \m@ne\ifx\MT@char\@empty\else\MT@get@char@unit\csname MT@feat@split@val\endcsname#2\relax\fi\fi\fi}
\MT@get@char@unit
\MT@pr@split@val#1,#2\relax{\def\@tempb{#1}}\%
Now we can set the values for the inheriting characters. Their slot numbers are
saved in the macro \MT@inh@〈list name〉@〈slot number〉.

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

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

The unit is in \MT@count, the desired factor in \@tempb, and the result will be
returned in \@tempcntb.
\MT@requires@pdftex3{\
\def\MT@scale@to@em{\
\@tempcntb=\MT@count\relax\
\MT@scale@factor}\}%
\MT@get@charwd

Get the width of the character. When using e-TeX, we can employ \fontcharwd
instead of building scratch boxes.
\def\MT@get@charwd{\
\ifnum\MT@char@<\z@\
\setbox\z@=\hbox{\MT@font \XeTeXglyph-\MT@char@}\
\else\
\ifnum\@tempcntb=\z@\else\
\setbox\z@=\hbox{\MT@font \char\MT@char}\%\
\fi\
\fi\
\MT@charcontains a slot number (legacy fonts), a Unicode number, or a glyph name
(if \MT@char@ is negative).
\MT@char
For letterspaced fonts, we have to subtract the letterspacing amount from the characters' widths. The protrusion amounts will be adjusted in \MT@set@pr@codes. The letterspaced font is already loaded so that 1em = \fontdimen 6.

No adjustment with versions 0.14f and 0.14g.

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

For the space unit.

Info about missing characters, or characters with zero width.

Furthermore, we might have to multiply with a factor.
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 and \SetTracking, which are being dealt with in \MT@get@ex@opt and \MT@get@tr@opt, resp.).

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

The codes are either relative to character widths, or to a fixed width. For spacing and kerning lists, they may also be relative to the width of the interword glue. Only the setting from the top list will be taken into account.
\let\MT@get@char@unit\relax
\let\MT@get@space@unit\@gobble
\MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
\let\MT@get@char@unit\MT@get@charwd
\else
\MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
\let\MT@get@space@unit\MT@get@font@dimen
\else
\MT@exp@cs\MT@get@unit{MT@\MT@feat @unit@}\
\fi
\fi

Preset all characters? If so, we surely don't need to reset, too.
\MT@ifdefined@n@T{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @preset}{%
\csname MT@preset@\MT@feat\endcsname
\MT@let@nc{MT@reset@\MT@feat @codes}\relax
%
}
\MT@get@unit
\MT@get@unit@

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

\MT@get@unit@{\def\MT@get@unit@#1{\expandafter\MT@get@unit@#1 e!\@nil\ifx\x\@empty\else\let#1\x\fi\@defaultunits\@tempdima#1 pt\relax\@nnil\ifdim\@tempdima=\z@\MT@warning@nl{\if\the\@tempdima\MT@vinfo{... : Setting \nameuse{MT@abbr@\MT@feat} factors relative to \the\@tempdima}\MT@count=\@tempdima\relax\fi\fi}}
\MT@get@unit@

\MT@set@inputenc

The configurations may be under the regime of an input encoding.
\MT@set@inputenc@% We remember the current category (c or inh), in case of warnings later.
\MT@set@inputenc@% More recent versions of inputenc remember the current encoding, so that we can test whether we really have to load the encoding file.
\MT@addto@setup{%\
\@ifpackageloaded{inputenc}{{\@ifpackagelater{inputenc}{2006/02/22}{%
\MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
\MT@load@inputenc
}%}
}\let\MT@set@inputenc@\MT@load@inputenc
%
\def\MT@set@inputenc@{%
\MT@warning@nl{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
\MessageBreak package isn't loaded. Ignoring input encoding}%
}
\MT@load@inputenc
Set up normal catcodes, since, e.g., listings would otherwise want to actually
typeset the \texttt{inputenc} file when it is being loaded inside a listing.
\def\MT@load@inputenc{%
\MT@cfg@catcodes
\debug\MT@dinfo@nl{1}{loading input encoding: \@nameuse{\@tempa}}%
\inputencoding{\@nameuse{\@tempa}}%
}\debug\MT@dinfo@nl{4}{;;; lp/rp (#1): \number\lpcode\MT@font\MT@char\space/%\number\rpcode\MT@font\MT@char\space}%
\MT@set@pr@heirs
Set the inheriting characters.
\MT@preset@pr
Preset characters. Presetting them relative to their widths is not allowed.
\MT@preset@aux
Auxiliary macro for presetting. Store value \texttt{(#1)} in macro \texttt{(#2)}. 

\MT@preset@aux@factor
\MT@preset@aux@space
\MT@warn\preset@towidth

14.2.2 Expansion

\MT@expansion

\MT@set@ex@codes@s

\MT@set@ex@codes@n

\ifMT@nonselected

\MT@set@ex@codes

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

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

\MT@reset@ef@codes

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

\MT@ex@split@val

There’s only one number per character.

\MT@scale

Take an optional factor into account.
Heirs, heirs, I love thy heirs.

\MT@ifdefined@c@T\ MT@ex@inh\ MT@name{Heirs\char\ MT@set@ex@heirs}\%
\MT@ifdefined@n@T{MT@inh\ MT@ex@inh\ MT@char\ MT@set@ex@heirs}\%
\MT@warn@too@large\%
\def\MT@warn@too@large#1{\MT@warning@nl{Expansion factor \number\@tempcntb\space too large for\MessageBreak\ the\MT@toks' in \MT@curr@list\MT@name.\MessageBreak\ Setting it to the maximum of \number#1}}\%
\@tempcntb=#1\relax

\MT@get@ex@opt \%
\def\MT@get@ex@opt{\MT@set@listname\%
\MT@ifdefined@n@TF{MT@ex@c@MT@ex@c@name \factor}{\MT@let@cn\MT@ex@factor@{MT@ex@c@MT@ex@c@name \factor}\MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}\%
\let\MT@ex@factor=\MT@ex@factor\%}
\MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}\%
\MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}\%
\MT@get@ex@opt@{step} {Setting expansion step to \number\MT@step@}\%
\MT@auto@\%
\MT@ifstreq{\MT@auto}{autoexpand}{En}{Dis}abling automatic expansion\%
\MT@preset@ex\%
\let\MT@reset@ef@codes\relax
}
\MT@get@ex@opt@\%
\def\MT@get@ex@opt@#1#2{\MT@ifdefined@n@TF{MT@ex@c@MT@ex@c@name #1}{\MT@let@nn{MT@#1@}{MT@ex@c@MT@ex@c@name #1}\MT@vinfo{... : #2}\%
\MT@let@nn{MT@#1@}{MT@#1}\%}

\MT@set@ex@heirs\%
\def\MT@set@ex@heirs#1{\efcode\MT@font#1=\efcode\MT@font\MT@char\%
\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@get@ex@opt0\%
\def\MT@get@ex@opt0#1#2#1\%
\MT@ifdefined@n@TF{MT@ex@c@MT@ex@c@name \#1}{\MT@let@mn{\MT@#1@}{\MT@ex@c@MT@ex@c@name \#1}\%
\MT@vinfo{... : \#2}\%
\MT@let@mn{\MT@#1@}{\MT@#1}\%
}

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

\MT@preset@ex\%
\def\MT@preset@ex{\@tempcntb=\csname MT@ex@0\endcsname\relax
\MT@scale@factor\%
\MT@set@all@ex\@tempcntb\%
}

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

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

\MT@get@ex@opt@\%
\def\MT@get@ex@opt@#1#2#1\%
\MT@ifdefined@n@TF{MT@ex@c@MT@ex@c@name \#1}{\MT@let@mn{\MT@#1@}{\MT@ex@c@MT@ex@c@name \#1}\%
\MT@vinfo{... : \#2}\%
\MT@let@mn{\MT@#1@}{\MT@#1}\%
}
14.2.3 Interword spacing (glue)

\MT@spacing

Adjustment of interword spacing? Only works with pdfTeX.

\MT@set@sp@codes

This is all the same.

\MT@sp@split@val

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

\MT@set@sp@heirs

\MT@reset@sp@codes
14.2.4 Additional kerning

\MT@kerning

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

\MT@requires{pdftex6}
\def\MT@kerning{\MT@maybe{kn}}
\MT@set@kn@codes

It's getting boring, I know.

\MT@set@kn@codes

Again, the unit may be measured in the space dimension; this time only \fontdimen 2.

\MT@kerning
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.

Tracking won’t work if the original font’s \fontdimen6 is zero, in which case we issue a warning (once for every font).

Zero tracking requires special treatment.

Letterspacing only works in PDF mode.

The letterspaced font instances are saved in macros \(\text{name}[/\text{letterspacing amount}]\).

In contrast to \MTfont, which may reflect the font characteristics more accurately (taking substitutions into account), \text{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.

\luatload provides the faux font feature kernfactor, which we will use when dealing with non-legacy fonts, as it is less problematic and faster than the pdfTEX primitive \letterspacefont.
Scale interword spacing (not configurable in \textit{letterspace}).

\begin{verbatim}
  \MAKEDEF{\textit{letterspace}}
  \MAKEDEF{\textit{letterspace}}\fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@ \relax sp
  \fontdimen2\MT@lsfont/1000\relax
  \end{verbatim}

Adjust outer kerning (\textit{microtype} only).

\begin{verbatim}
  \MAKEDEF{\textit{letterspace}}
  \ MAKEDEF{\textit{letterspace}}\tagcode\MT@lsfont`f=\m@ne
  \end{verbatim}

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

\begin{verbatim}
  \MAKEDEF{\textit{letterspace}}\\MT@font\aftergroup\MT@set@lsfont
  \aftergroup\MT@set@curr@ls
  \end{verbatim}

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

\begin{verbatim}
\end{verbatim}

We need to remember the current letterspacing amount (for \texttt{\lslig}).

\begin{verbatim}
\end{verbatim}

Adjust surrounding spacing and kerning.

\begin{verbatim}
\end{verbatim}

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

\ifx\MT@ls@adjust\@empty
\MT@outer@kern=-\dimexpr\MT@letterspace@ sp \fontdimen6\font@name/2000\relax
\else
\ifdim\MT@outer@kern=0\relax
\MT@ls@outer@k
\else
\MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo
\csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
\ifdim\MT@outer@kern=0\relax
\MT@ls@outer@k
\fi
\MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo
\csname MT@outer@kern\expandafter\string\font@name\endcsname\relax
\fi
\fi
\fi
\MT@afteraftergroup{\
\MT@set@curr@ok\
\MT@set@curr@os\
\MT@tr@outer@r}
\fi
\fi
\MT@afteraftergroup

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

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

\def\MT@ls@fontspec#1:#2:#3:#4\@nil{% \ifx\#3\#1:#2\else#1:#2:#3\fi}

\def\MT@ls@fontspec@colon#1:#2:#3:#4\@nil{% \ifnum\MT@minus\MT@letterspace<1000 1\else 0\fi \ifnum\MT@minus\MT@letterspace<10 0\fi \ifnum\MT@minus\MT@letterspace<1 \MT@letterspace\fi;}

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

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

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

\begin{verbatim}
\DeclareRobustCommand\lsstyle{\not@math@alphabet\lsstyle\textls\let\glb@currsize\@empty}〈pdftex-def|luatex-def〉\MT@maybe@gobble@with@tikz\aftergroup\glb@settings｝〈pdftex-def|luatex-def〉\def\MT@feat{tr}\let\MT@tracking\MT@set@tr@codes\selectfont\end{verbatim}

Now the definitions for the letterspace package with plain TEX.

\begin{verbatim}
\def\MT@set@lsfont\MT@lsfont\def\lsstyle{\begingroup\escapechar\m@ne\xdef\font@name{\csname\expandafter\string\the\font\endcsname}\MT@set@tr@codes\endgroup}\let\textls\@undefined\let\lslig\@undefined\end{verbatim}

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

\begin{verbatim}
\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\MT@dinfo@nl{1}{... fixing base font}\MT@exp@two@c\let\font@name\MT@ls@basefont\MT@dinfo@nl{1}{}}\MT@ls@basefont\MT@get@ls@basefont\pdfTEX cannot letterspace fonts that already are letterspaced. Therefore, we have to save the base font in \font name@base. The previous solution (checking the macro's meaning with \pdfmatch), where we were loading the base font via the \font primitive again, would destroy all previously set up micro-typographic features of the font.
\end{verbatim}

\begin{verbatim}
\MT@ls@basefont\MT@get@ls@basefont\MT@set@lsbasefont\MT@set@tr@zero\MT@set@tr@zero If tracking is switched off in the middle of the document, or if \textls is called with a zero letterspacing amount, we have to retrieve the base font and select it.
\end{verbatim}
\def\MT@tr@noligatures{
\ifx\MT@tr@ligatures\@empty
\MT@noligatures@\MT@lsfont\@undefined
\else
\MT@noligatures@\MT@lsfont\MT@tr@ligatures
\fi}
\def\MT@tr@set@space#1,#2,#3,#4,#5,#6,\{
\MT@ifempty{#2}{\MT@ifempty{#1}\relax{\MT@tr@set@space@@{#1}{#3}{2000}}\fontdimen#3\MT@lsfont=\@tempdima}{\MT@tr@set@space@@{#2}{#3}{2000}\edef\MT@temp{\MT@temp#4\the\@tempdima}\MT@ifempty{#1}\relax{\MT@tr@set@space@@{#1}{#3}{2000}}}
\MT@tr@set@space@@{#2}{#3}{2000}\edef\MT@temp{\MT@temp#4\the\@tempdima}\MT@ifempty{#1}\relax{\MT@tr@set@space@@{#1}{#3}{2000}}}
\MT@outer@space A new skip for outer spacing.
\MT@outer@space
\MT@tr@set@space Adjust interword spacing (\fontdimen2,3,4) 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@space If settings for outer spacing (#2) don’t exist, they will be inherited from the inner spacing settings (#1).
\MT@tr@set@space
If the value is followed by an asterisk, the fontdimen will be scaled by the respective amount, otherwise the value denotes the desired dimension in the respective unit.

```
def\MT@tr@set@space@@#1#2#3{\
  \MT@test@ast#1*\@nil{\
    \MT@ifdefined@c@TF\MT@tr@unit@\
    {\edef\@tempb{#1}\MT@scale@to@em}\
    {\@tempcntb=#1\relax}\
    \@tempdima=\dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax\
    \ifnum#2=\tw@\
    \advance\@tempdima -\dimexpr\MT@letterspace sp*\MT@dimen@six/#3\relax\
    \fi\
  }{\
    \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax\
    \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax\
  }\
  \MT@dinfo@nl3{... : font dimen #2 (#1): \the\@tempdima}}\
```

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

```
\MT@tr@outer@l\
\MT@tr@outer@r\
```

microtype also adjusts spacing. The following is borrowed from soul. I've added the cases for italic correction, since tracking may also be triggered by text commands.
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).

xspace requires special treatment.
If there's no outer spacing, there may be outer kerning.
\def\MT@temp*\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k\fi\fi
\MT@let@nc{MT@tr@outer@next}\relax
\fi\fi
\MT@temp*
\MT@tr@outer@icr
\MT@tr@outer@icr@
Helper macros for the italic correction mess.
\def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
\def\MT@tr@outer@icr@{
\let\@let@token=\MT@tr@outer@next
\maybe@ic@
}\MT@xspace\MT@xspace@
If the group is followed by \xspace, we first feed \xspace with the next token, then check whether it has inserted a space. \@let@token might be something evil, so it should be encapsulated here.
\def\MT@xspace{\futurelet\@let@token\MT@xspace@}
\def\MT@xspace@{\@xspace@firsttrue\@xspace\ifdim\lastskip>5sp\unskip\hskip\MT@outer@space\else\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k\fi\fi}
For older pdf\TeX\ versions and Lua\TeX, throw an error.
\DeclareRobustCommand\lsstyle{\MT@error{Letterspacing only works with \MT@engine\ tex version \pdftex-def 1.40\luatex-def 0.62\MessageBreak or newer}{Upgrade \MT@engine\ tex, or try the `soul' package instead.}}
\MT@glet\lsstyle\relax
And for X\TeX, too.
\DeclareRobustCommand\lsstyle{\MT@error{Letterspacing currently doesn't work with xetex}{Run pdftex or luatex, or use the `soul' package instead.}}
\MT@glet\lsstyle\relax
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.
\textls This is now almost \LaTeX's \DeclareTextFontCommand, with the difference that we
adjust the outer spacing and kerning also for \lsstyle, while \TeX's text switches don't bother about italic correction.

\newcommand\MT@ls@set@ls[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@adjust
Set current letterspacing amount and outer kerning. This has to be done inside the same group as the letterspacing command.

\MT@ls@adjust@empty
\MT@ls@adjust@relax
\MT@ls@adjust@empty
\MT@ls@adjust@relax
\MT@ls@too@large
Test whether letterspacing amount is too large.

\MT@outer@kern
This dimen is used for the starred version of \textls, for \lslig and for adjusted outer kerning.
Adjust outer kerning. We additionally add a marker (\kern3sp\kern-3sp) for cases of nested letterspacing without anything actually printed.

The possibility to disable ligatures is a new feature of pdfTeX 1.30, and also works with LuaTeX.

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

No 'inputenc' key.

Early MiKTeX versions (before 2.5.2579) didn’t know \tagcode.
With LuaTeX, we additionally register the ligatures that should be inhibited in a table (used by the luaotfload function keepligature).

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

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

```lua
if luaotfload and luaotfload.letterspace then
  luaotfload.letterspace.keepligature = keepligature
end
```

14.2.7 Loading the configuration

\MT@load@list

\MT@find@file

Micro-typographic settings may be written into a file $\texttt{mt-}<$font family$>.cfg$.

We must also record whether we've already loaded the file.

\MT@find@file

Check for existence of the file only once.

[package]
\def\MT@load@list\{}% 
\def\@empty\{}% 
\def\MT@find@file#1{% 
\InputIfFileExists{mt-#1.cfg}{
\edef\MT@curr@file{mt-#1.cfg} 
\MT@vinfo{... Loading configuration file \MT@curr@file} 
\MT@xadd\MT@file@list{#1,} %
}% 
\else 
\MT@get@basefamily#1\@empty\@empty\@empty\@nil 
\MT@exp@one@n\MT@in@clist\@tempa\MT@file@list 
\ifMT@inlist@ 
\MT@xadd\MT@file@list{#1,} 
\else 
\InputIfFileExists{mt-\@tempa.cfg}{
\edef\MT@curr@file{mt-\@tempa.cfg} 
\MT@vinfo{... Loading configuration file \MT@curr@file} 
\MT@xadd\MT@file@list{\@tempa,#1,} 
\MT@error{\@nameuse{MT@abbr\@MT@feat} list `\@tempb' undefined. \MessageBreak Cannot load it from list `\@tempa'}{}% 
}%
\fi 
}%

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 tabulars), !, ?, ; (french), $, _, ~, and = (Turkish babel).

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

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

\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.

\MT@get@basefamily0

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

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

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.

The current context is added to the font attributes. That is, the context must match.

Also try with an alias family.
\MT@context% \debug\MT@info@nl{1}{(alias) \tempa}% \MT@ifdefefined\MT@tempb \@tempa{% \MT@next@listname@#4% \if \fi \fi % }\MT@next@listname@

If size is to be evaluated, do that, otherwise use the current list.
\def\MT@next@listname@#1{% \ifnum#1=1 \MT@exp@cs\MT@in@rlist{MT@\@tempb \@tempa @sizes}% \ifMT@inlist@
 \let\MT@listname\MT@size@name \fi \else \MT@let@cn\MT@listname{MT@\@tempb \@tempa} \fi % \MT@if@list@exists \MT@context \def\MT@if@list@exists{% \MT@let@cn\MT@context{MT@\MT@feat @context} \MT@ifstreq{@}\MT@context{@empty}\relax \MT@get@listname{\MT@feat @c} \MT@ifdefined@c@TF\MT@listname{% \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname} \ifMT@nonselected \MT@vinfo{... Applying non-selected expansion (list `\MT@listname')} \else \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\MT@listname'} \fi \@secondoftwo \MT@let@nc{MT@\MT@feat @c@name}@empty \MT@let@cn\MT@context{MT@\MT@feat @c@name}@empty \MT@let@nc{MT@\MT@context}@empty \ifMT@nonselected \MT@vinfo{... Applying non-selected expansion (no list)} \else Tracking doesn't require a list, either. \MT@fistreq\MT@feat{tr}\relax \MT@Warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list for font `\MT@context' \ifx\MT@context@empty\space(context: `\MT@context')\fi. Switching off `\MT@context'
\@nameuse{MT@abbr@\MT@feat} for this font} \fi \fi \@secondoftwo \MT@get@inh@list \MT@context \def\MT@get@inh@list{% \let\MT@context@empty \MT@get@listname{\MT@feat @inh} \MT@ifdefined@c@TF\MT@listname{% \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname} % Since the name cannot be \@empty, this is a sound proof that no matching list
exists. \MT@let@nc{MT@\MT@feat @c@name}@empty \MT@let@cn\MT@context{MT@\MT@feat @c@name}@empty \ifMT@nonselected \MT@vinfo{... Applying non-selected expansion (no list)} \else Tracking doesn't require a list, either. \MT@fistreq\MT@feat{tr}\relax \MT@Warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list for font `\MT@context' \ifx\MT@context@empty\space(context: `\MT@context')\fi. Switching off `\MT@context'
\@nameuse{MT@abbr@\MT@feat} for this font} \fi \fi \@secondoftwo \MT@get@inh@list \MT@context \def\MT@get@inh@list{% \let\MT@context@empty \MT@get@listname{MT@\MT@feat @inh} \MT@ifdefined@c@TF\MT@listname{% \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname} % The inheritance lists are global (no context). \MT@context \def\MT@context{% \MT@get@listname{MT@\MT@feat @inh} \MT@ifdefined@c@TF\MT@listname{% \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
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@char

\MT@char0

\MT@get@slot

\MT@char

\MT@char0

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

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

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

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

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

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

  If \(\langle\text{encoding}\rangle\backslash\langle\text{command}\rangle\) (that's one command) is defined, we try to extract the slot number.

  We must be cautious not to stumble over accented characters consisting of two commands, like \`\i or \CYRI, hence, \string wouldn't be safe enough.
• Now, we'll catch the rest, which hopefully is an accented character (e.g. "á").

```latex
\begin{verbatim}
\ifnum\MT@char\ < \z@ \fi
\fi
\fi
\let\MT@char\MT@char@
\MT@get@slot@
\escapechar\m@ne
\end\verbatim
```

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

```latex
\begin{verbatim}
\expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
\meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
\fi
\fi
\let\MT@char\MT@char@
\MT@get@slot@
\escapechar\m@ne
\end{verbatim}
```

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.

```latex
\begin{verbatim}
\if\MT@norest\else
\MT@warn@\<\MT@char\>
\fi
\fi
\else
\MT@warn@\<\MT@char\>
\fi
\end{verbatim}
```

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

\ifnum\MT@char=47\relax
  \ifMT@norest \edef\MT@char{U47}\% 
  \else
    \tempcnta=\XeTeXglyphindex\expandafter\@gobble\tempa\relax
    \ifnum\tempcnta=0
      \MT@warn@Unknown
      \let\MT@char=\empty
    \else
      \edef\MT@char{\tempa}\space
      \edef\MT@char@{-\the\tempcnta}\% 
      \\MT@dinfo@nl{3}{> \the\MT@toks is a glyph name (\the\tempcnta)}\%
    \fi
  \fi
\else
  \ifnum\MT@char > -1
    \ifMT@norest
      Or, it's a Unicode number, which we mustn't translate into a glyph number, since the latter is font-specific.
      \tempcnta=\XeTeXcharglyph\MT@char\relax
      \ifnum\tempcnta=0
        \MT@warn@MissingChar
        \let\MT@char=\empty
      \else
        \\debug\MT@dinfo@nl{3}{> (glyph number: \the\tempcnta, glyph name: \XeTeXglyphname\MT@font\tempcnta)}\%
        \edef\MT@char{U\MT@char}\%
      \fi
    \else
      \MT@warn@Rest
      \let\MT@char=\empty
    \fi
  \else
    \MT@warn@Unknown
    \let\MT@char=\empty
  \fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi

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.

\ifluafile
  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) return slot_of_name(font.current(), name, unsafe) end
  end
\else
  -- legacy luaotfload
  local tfmdata = fonts.ids[font.current()]
  if not tfmdata then return end
  unicodes = tfmdata.shared.otfdata.luatex.unicodes
  if fonts.ids then
    local unicodes = fonts.ids[font.current()]
    if not unicodes then return end
  end
  local tfmdata = fonts.hashes.identifiers[font.current()]
  if not tfmdata then return end
\fi
unichores = tfmdata.resources.unichores
end
local unichore = unichores[name]
if unichore then -- does the 'or' branch actually exist?
    return type(unichore) == "number" and unichore or unichore[1]
end
end
end

microtype.name_to_slot = name_to_slot
end

〈/luafile

\MT@is@letter
Input is a letter, a character or a number.
\MT@max@slot
Warning if resulting character or slot number is too large.
\MT@max@char
\MT@max@char
\MT@max@slot
\MT@max@slot
\MT@is@number
Numbers may be specified as a three-digit decimal number (029), as a hexadecimal
number (prefixed with "\": 1D) or as a octal number (prefixed with '\": 35). They
must consist of at least three characters (including the prefix), that is, "F is not
permitted.
Expand an active character. (This was completely broken in v1.7, and only worked by chance before.) We set@display@protect to translate, e.g., Ä into "A, that is to whatever it is defined in the inputenc encoding file.

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

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

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

For ucs (utf8x). Let’s call it experimental ...

The \expandafter hocus-pocus should please newunicodechar.

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

For characters not defined in the current input encoding.

The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding \langle command⟩, we construct the command \langle encoding⟩\langle command⟩.
and see whether its meaning is \char"\{hex number\}, which is the case for everything that has been defined with \DeclareTextSymbol in the encoding definition files.

For TU encoding, the commands \textquotesingle, \textasciigrave and \textquotedbl are defined by means of the auxiliary macro \remove@tlig, which we take care of here.

Finally, if it hasn't been defined by \DeclareTextSymbol, it could be a letter (e.g., \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.
This might have to change again with the next \LaTeX{} release, ... or so I feared, but it still seems to be fine.

\def\MT@is@tlig#1#2{\ifx#1\remove@tlig\debug\MT@dinfo@nl{3}{> \the\MT@toks' (removing remove@tlig)}\fi\MT@is@number #2\relax\relax\fi}

Here, we are dealing with accented characters, specified as two tokens.

\def\MT@is@composite#1#2\relax{\ifx\\else\Again, we construct a control sequence, this time of the form: \langle\langle\encoding\langle\accent\langle\character\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\ratio
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 \textit{cmsy} when the \textit{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.

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

Number too large.

Not all of the string has been parsed.

No idea what went wrong.

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
\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@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 unicode-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}}%
font | \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. \% 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. \% This is the normal \LaTeX{} definition. Check whether \pickup@font is defined as expected. The warning issued by \CheckCommand* would be a bit too generic. \% Then we append our stuff. Everything is done inside a group.
If the `trace` package is loaded, we turn off tracing of `microtype`'s setup, which is extremely noisy.

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.

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.

\MT@registerfont \ Register the current font.
\MT@register@font{\def\MT@registerfont{\xdef\MTfont@list{\MTfont@list\MTfont}}}

\MT@register@substfont \ Register the substituted font (only if it isn't registered already). Additionally, we have to remove the substitute font from the list of fonts, so that we set it up again.
\MT@register@substfont{\def\MT@register@substfont{% \MT@register@substfont{\MT@exp@one@n\MT@in@clist\font@name\MTfont@list} \if\MT@inlist@ \else \xdef\MTfont@list{\MTfont@list\font@name,} \fi \expandafter\MT@rem@from@clist\MTfont\MTfont@list \fi \}}

\MT@active@features \ The activated features are stored in this command.
\MT@active@features{\let\MT@active@features\@empty}

\MT@check@font@cx \ Every feature has its own list of fonts that have already been dealt with. If the font needn’t be set up for a feature, we temporarily disable the corresponding setup command. This should be more efficient than book-keeping the fonts in lists associated with the combination of contexts, as we’ve done it before.
\MT@check@font@cx{\def\MT@check@font@cx{% \if\MT@if@true \MT@map@clist@c\MT@active@features{% \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname \if\MT@inlist@ \else \MT@exp@cs\MT@xadd{MT@##1@\csname MT@##1@context\endcsname font@list}{\font@name,} \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MTfont\csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname \fi }% \if\MT@if@ \MT@inlist@true \else \MT@inlist@false \fi }}

\MT@register@substfont@cx \ Add the substituted font to each feature list and possibly remove substitute font.
\MT@register@substfont@cx{\def\MT@register@substfont@cx{% \MT@register@substfont@cx{% \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname \if\MT@inlist@ \else \MT@exp@cs\MT@xadd{\csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname}{\font@name,} \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MTfont\csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname \fi }}

\MT@register@font@cx \ For each feature, add the current font to the list, unless we didn't set it up.
\MT@register@font@cx{\def\MT@register@font@cx{% \MT@register@font@cx{% \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname \if\MT@inlist@ \else \MT@exp@cs\fx{MT@\Onameuse{MT@abbr@##1}}\relax\else \MT@exp@cs\MT@xadd \fi }}
\MT@maybe@rem@from@list
Recurse through all context font lists of the document and remove the font, unless it's the current context.
\def\MT@maybe@rem@from@list#1{%\ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%\expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font\csname MT@\@tempa @#1font@list\endcsname}%}%
\textmicrotypecontext
This is just a wrapper around \microtypecontext.
\def\textmicrotypecontext{%\MT@begin@catcodes\MT@textmicrotypecontext\selectfont\MT@reset@context\MT@reset@context@}%
\MT@setup@contexts
The first time \microtypecontext is called, we initialise the context lists and redefine the commands used in \pickup@font.
\def\MT@setup@contexts{%\MT@map@clist@c\MT@active@features{%\MT@glet@nc{MT@##1@@font@list}\MT@font@list%\MT@glet\MT@check@font\MT@check@font@cx\MT@glet\MT@register@font\MT@register@font@cx}%\MT@maybe@rem@from@list}{\MT@maybe@rem@from@list}{%\MT@begin@catcodes\MT@microtypecontext\selectfont\MT@reset@context\MT@reset@context@}%
\MT@setup@contexts
The user may change the context, so that different setups are possible. This is especially useful for multi-lingual documents.
Inside the preamble, this command shouldn't actually do anything but remember itself for later.
\def\microtypecontext{%\MT@begin@catcodes\MT@microtypecontext\selectfont}%
\MT@glet\glb@currsize\@empty\selectfont\MT@reset@context\MT@reset@context@%
Define context keys.

Using an empty context is only asking for trouble, therefore we choose the '∅' instead (hoping for the \LaTeX\ users' natural awe of this character).

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

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

We also allow the activate shortcut.

Initialise the contexts.

Configuration

Font sets

Calling this macro will create a comma list for every font attribute of the form \texttt{\MT\texttt{\{feature\}}}\texttt{\{list\}}\texttt{\{attribute\}}\texttt{\{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.
\begin{verbatim}
\MT@begin@catcodes
\@ifstar
\MT@DeclareSetAndUseIt
\MT@DeclareSet
\MT@DeclareSetAndUseIt
\newcommand\MT@DeclareSetAndUseIt[3][]{
\MT@DeclareSet[#1]{#2}{#3}
\UseMicrotypeSet[#1]{#2}
}
\MT@curr@set@name
We need to remember the name of the set currently being declared.
\let\MT@curr@set@name\@empty
\MT@declare@sets
\def\MT@declare@sets#1#2#3{%
\def\MT@curr@set@name{#2}%
\MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
\MT@warning{Redefining \@nameuse{MT@abbr@#1} set \`\MT@curr@set@name'}%
\MT@map@clist@n{font,encoding,family,series,shape,size}{%
\MT@glet@nc{MT@#1list@##1@\MT@curr@set@name}@\@undefined%
}}%
\MT@glet@nc{MT@#1@set@@\MT@curr@set@name}@\@empty
\endverbatim
Saying, for instance, `family=rm*` 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 versions, these values will not be expanded immediately, but at the end of the preamble.

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 \(-1\).

\begin{verbatim}
\def\MT@get@range{\if\relax#1\else\MT@ifempty{#1}{% \ MT@ifempty{#2}{% \let\MT@val\relax % \MT@get@size % \edef\MT@upper{\MT@val} % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % } % \MT@ifdim\MT@lower>\MT@val{\MT@error{\textbf{Invalid size range (\MT@lower>\MT@val) in font set \`\MT@curr@set@name'.}}% % \edef\MT@upper{\MT@lower} % \edef\MT@lower{\MT@val} % }% % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % \relax % }% % \MT@ifempty{#2}{% % \MT@ifempty{#3}{% % {\def\MT@upper{-1}}% % }% % \def\MT@val{#2} % \MT@get@size % \ifx\MT@val\relax% % \MT@ifdim\MT@lower>\MT@val{\MT@error{\textbf{Invalid size range (\MT@lower>\MT@val) in font set \`\MT@curr@set@name'.}}% % \edef\MT@upper{\MT@lower} % \edef\MT@lower{\MT@val} % }% % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % \relax % }% % \def\MT@val{#1} % \MT@get@size % \ifx\MT@val\relax% % \MT@ifdim\MT@lower>\MT@val{\MT@error{\textbf{Invalid size range (\MT@lower>\MT@val) in font set \`\MT@curr@set@name'.}}% % \edef\MT@upper{\MT@lower} % \edef\MT@lower{\MT@val} % }% % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % \relax % }% % \MT@ifempty{#3}{% % {\def\MT@upper{-1}}% % }% % \def\MT@val{#2} % \MT@get@size % \ifx\MT@val\relax% % \MT@ifdim\MT@lower>\MT@val{\MT@error{\textbf{Invalid size range (\MT@lower>\MT@val) in font set \`\MT@curr@set@name'.}}% % \edef\MT@upper{\MT@lower} % \edef\MT@lower{\MT@val} % }% % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % \relax % }% % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % \relax % }% % \MT@ifdim\MT@lower=\MT@upper % {\def\MT@upper{-1}}% % \relax % }% % % % \MT@get@size % \if*\MT@val\relax% % \def\@tempa{\normalsize} % \else % \MT@let@cn\@tempa{\MT@val} % \fi % \ifx\@tempa\relax% % % % \else % \relsize % \endgroup % \begin{group}
\def\MT@get@size潸 % Translate a size selection command and normalise it.
\end{verbatim}

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

\begin{verbatim}
\def\MT@get@size潸 % \begin{group}
\def\set@fontsize##1##2##3##4\@nil{
\MT@get@size % \end{group}
\end{verbatim}

The \texttt{\relsize} solution of parsing \texttt{\setfontsize} does not work with the \texttt{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).
Test whether we finally got a number or dimension so that we can strip the ‘pt’ (\@defaultunits and \strip@pt are kernel macros).
\MT@ifdimen\MT@val{% 
\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 
〈debug〉\MT@dinfo@nl{1}{-- font: \@nameuse{MT@#1list@font@\MT@curr@set@name}}\% 
\}{} 
}{} 
\MT@get@font 
\MT@get@font 
\MT@get@font@ 
\MT@get@font@#1#2#3#4#5#6{% 
\let\@tempb\@empty 
\def\MT@val{#2}{} 
\MT@get@highlevel{#1}{} 
\MT@ifempty\MT@val{}{\MT@warn@axis@empty{size}{\string
\normalsize}}{} 
\def\MT@val{#5}{} 
\MT@get@size
We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{% 
\MT@define@set@key{encoding}{#1}% 
\MT@define@set@key{family}{#1}% 
\MT@define@set@key{series}{#1}% 
\MT@define@set@key{shape}{#1}% 
\MT@define@set@key{size}{#1}% 
\MT@define@set@key{font}{#1}% }

\UseMicrotypeSet

To use a particular set we simply redefine MT@〈feature〉setname. If the optional argument is empty, set names for all features will be redefined.

\def\UseMicrotypeSet{% 
\MT@begin@catcodes 
\MT@UseMicrotypeSet 
}\MT@UseMicrotypeSet

\newcommand\MT@UseMicrotypeSet[2][]{% 
\MT@ifempty{#1}{% 
\MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}% }{% 
\MT@map@clist@n{#1}{{% 
\MT@ifempty{##1}\relax{% 
\MT@is@feature{##1}{activation of set `#2'}{% 
\MT@exp@one@n\MT@use@set \csname MT@rbba@##1\endcsname}{#2}% }% 
}}% 
}% 
}\MT@end@catcodes

\def\MT@use@set#1#2{% 
\MT@ifdefined@n@TF{MT@#1@set@@#2}{}{% 
\MT@xdef@n{MT@#1@setname}{#2}% 
}\MT@ifdefined@n@TF{MT@#1@setname}\relax{% 
\MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}% 
}\MT@error{% 
The \@nameuse{MT@abbr@#1} set `#2' is undeclared. Using set `\@nameuse{MT@#1@setname}' instead}{} 
}%

\DeclareMicrotypeSetDefault

This command can be used in the main configuration file to declare the default font set, in case no set is specified in the package options.
\DeclareMicrotypeVariants
\MTvariants
Specify suffixes for variants (see fontname/variants.map). The starred version appends to the list.

\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.
\MTDeclareMicrotypeAlias

\newcommand*{\MTDeclareMicrotypeAlias}[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}%

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

\MT@ifdefined@c@T\MT@family{%
\debug\MT@dinfo{1}{Activating alias font `\@tempb' for `\MT@family'}%
\MT@glet\MT@familyalias\@tempb
\MT@end@catcodes%
}%
\MT@begin@catcodes
\LoadMicrotypeFile May be used to load a configuration file manually.
\def\LoadMicrotypeFile#1{%
\edef\@tempa{\zap@space#1 \@empty}%
\@onelevel@sanitize\@tempa
\MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
\ifMT@inlist@
\MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
\else
\MT@xadd\MT@file@list{\@tempa,}%
\MT@begin@catcodes
\InputIfFileExists{mt-\@tempa.cfg}{%
\edef\MT@curr@file{mt-\@tempa.cfg}%
\MT@vinfo{... Loading configuration file \MT@curr@file}%
}%
\MT@warning{Configuration file mt-\@tempa.cfg does not exist}%
}%
\MT@end@catcodes%
}}%
\MT@end@catcodes%
}%

\LoadMicrotypeFile

14.3.3 Disabling ligatures
\DisableLigatures
\MTDisableLigatures
\MTon@setname

\MTon@ligatures

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

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

If pdftex is too old, we throw an error.

\renewcommand*=\DisableLigatures{\}{% 
\MT@error{Disabling ligatures of a font is only possible with pdftex version 1.30 or newer.\MessageBreak ignoring \string\DisableLigatures}{% 
Upgrade\pdftex.}%
%
}

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.

\SetExpansion
\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 \#3 (i.e., the list of characters, not expanded).

\MT@SetProtrusion

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

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

Parse the optional first argument. We first have to know the name before we can deal with the extra options.

\MT@set@named@keys{MT@pr@c}{#1}%
\MT@permute

We have parsed the second argument, and can now define macros for all permutations of the font attributes to point to \MT@pr@c@, . . .
... which we can now define to be \(#3\). Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

\begin{verbatim}
\MT@edef\MT@pr@c@\MT@pr@c@name {#3}\
\MT@end@catcodes
\end{verbatim}

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

\begin{verbatim}
\MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}\
\MT@end@catcodes
\end{verbatim}

\SetTracking
\begin{verbatim}
\MT@gdef@n{MT@tr@c@\MT@tr@c@name}{#3}\
\MT@end@catcodes
\end{verbatim}

\SetExtraSpacing
\begin{verbatim}
\MT@gdef@n{MT@ex@c@\MT@ex@c@name}{#3}\
\MT@end@catcodes
\end{verbatim}
\MT@SetExtraSpacing
\def\MT@SetExtraSpacing[#1][#2][#3]{{\let\MT@extra@context\@empty\MT@set@named@keys{MT@sp@c}{#1}{\MT@dinfo{1}{creating spacing list \texttt{\MT@sp@c@name'}}\setkeys{MT@cfg}{#2}{\MT@permute\def\MT@permutelist{sp@c}\MT@gdef@n{MT@sp@c@MT@sp@c@name}{#3}{\MT@end@catcodes}}}}

\SetExtraKerning
\def\SetExtraKerning{{\MT@begin@catcodes\MT@SetExtraKerning\MT@end@catcodes}}

\MT@SetExtraKerning
\def\MT@SetExtraKerning[#1][#2][#3]{{\let\MT@extra@context\@empty\MT@set@named@keys{MT@kn@c}{#1}{\MT@dinfo{1}{creating kerning list \texttt{\MT@kn@c@name'}}\def\MT@permutelist{kn@c}\setkeys{MT@cfg}{#2}{\MT@permute\def\MT@permutelist{kn@c}\MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}{\MT@end@catcodes}}}}

\MT@set@named@keys
We first set the name (if specified), then remove it from the list, and set the remaining keys.
\def\MT@set@named@keys#1#2{\def\x##1name=##2,##3\@nil{\setkeys{#1}{name=##2}{\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=\numexpr\the\@tempcnta+1\relax\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\@tempcnta1}}}}

\MT@define@code@key@family
Remove fontspec's internal feature counter.
\def\MT@define@code@key@family#1{\def\MT@define@code@key#1\@nil\@expandtwoargs\setkeys{#1}{\MT@options}}
\define@key{MT@#1}{family}[]{\%}
\@tempcnta=\@ne
\MT@map@clist@n{##1}{%}
\KV@@sp@def\MT@val{####1}%%
\MT@get@highlevel{family}%%
\ifMT@fontspec
\edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
\fi
\MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}%%
\advance\@tempcnta \@ne
%
\MT@define@code@key@size
\MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.
\def\MT@define@code@key@size#1{%
\define@key{MT@#1}{size}[]{%}
\MT@map@clist@n{##1}{%}
\KV@@sp@def\MT@val{####1}%%
\expandafter\MT@get@range\MT@val--\@nil
\ifx\MT@val\relax\else
\MT@exp@cs\MT@xadd{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}%
\fi
\}}
%
\MT@define@code@font
\def\MT@define@code@key@font#1{%
\define@key{MT@#1}{font}[]{%}
\MT@map@clist@n{##1}{%}
\KV@@sp@def\MT@val{####1}%%
\MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
\expandafter\MT@get@font@and@size\MT@val/////\@nil
\ifMT@fontspec
\edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}\
\fi
\MT@xdef@n{MT@\MT@permutelist@\@tempb\MT@extra@context}\
\csname MT@\MT@permutelist@name\endcsname}\
〈debug〉\MT@dinfo@nl{1}{initialising: use list for font \@tempb=\MT@val
〈debug〉\ifx\MT@extra@context\@empty\else\MessageBreak
〈debug〉(context: \MT@extra@context)\fi}\
\MT@exp@cs\MT@xaddb{{\MT@val}{\m@ne}{\MT@curr@set@name}}%
\}}
%
\MT@get@font@and@size
Translate any asterisks and split off the size.
\def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
\MT@get@font{#1}{#2}{#3}{#4}{#5}{1}%
%
\MT@define@opt@key
\def\MT@define@opt@key#1#2{%
\define@key{MT@#1@c}{#2}{
\MT@ifempty{##1}\relax{%

}\}}
%
\MT@define@code@key{encoding}{cfg}
\MT@define@code@key@family {cfg}
\MT@define@code@key{series} {cfg}
\MT@define@code@key{shape} {cfg}
\MT@define@code@key@size {cfg}
\MT@define@code@key@font {cfg}
\MT@define@opt@key
\def\MT@define@opt@key#1#2{%
\define@key{MT@#1@c}{#2}{}
The options in the optional first argument.

Use file name and line number as the list name if the user didn’t bother to invent one – also check whether the name already exists (in case more than one unnamed list is loaded in the same line, for example \AtBeginDocument).

Only one context is allowed. This might change in the future.

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdf\LaTeX\ versions, disallow. It also works with Lua\LaTeX\ 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 pdf\LaTeX\ 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.

\begin{verbatim}
\define@key{MT@pr@c}{context}{\relax
\MT@ifempty{#1}\relax{\MT@glet\MT@copy@font\MT@copy@font@
\def\MT@extra@context{#1}}}
\MT@addto@setup{%\define@key{MT@pr@c}{context}{\relax
\MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}}{
\ifx\MT@copy@font\MT@copy@font@\else
\MT@warning@nl{If protrusion contexts don’t work as expected,
\MessageBreak load the package with the `copyfonts' option}{}%}
\fi}
\end{verbatim}

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

\begin{verbatim}
\define@key{MT@pr@c}{unit}{character
\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}@empty
\def\@tempa{#1}
\MT@ifstreq\@tempa{character}\relax{\Test whether it’s a dimension, but do not translate it into its final form here, since
it may be font-specific.
\MT@fifdimen\@tempa
\{\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}@empty\}
\def\@tempa{#1}%
\MT@fif@streq\@tempa{character}\relax{\Test whether it’s a dimension, but do not translate it into its final form here, since
it may be font-specific.
\MT@fifdimen\@tempa
\{\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}@empty\}
\def\@tempa{#1}%
\MT@warn@nodim}\end{verbatim}

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.

A space terminates the number.

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).

\MT@define@opt@key{tr}{spacing}
\MT@define@opt@key{tr}{outerspacing}
\MT@define@opt@key{tr}{outerkerning}

Which ligatures should be disabled?
\define@key{MT@tr@c}{noligatures}
\define@key{MT@tr@c}{outer spacing}
\define@key{MT@tr@c}{outer kerning}
\define@key{MT@tr@c}{no ligatures}

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

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

\MT@set@inh@feat
\MT@extra@inputenc

\DeclareCharacterInheritance{package}
\renewcommand\DeclareCharacterInheritance[1]{
\let\MT@extra@context\@empty
\let\MT@extra@inputenc\@undefined
\let\MT@inh@feat\@empty
\setkeys{MT@inh@}{#1}%
\MT@begin@catcodes
\MT@set@inh@list
\MT@end@catcodes}

\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@declare@char@inh
\MT@map@clist@c\MT@features{\MT@declare@char@inh}{#1}{#2}{\MT@declare@char@inh}{#1}{#2}}
\MT@map@clist@c\MT@features{\KV@@sp@def\@tempa{##1}
\MT@ifempty\@tempa\relax{\MT@exp@one@n\MT@declare@char@inh
\csname MT@rbba@\@tempa\endcsname}{#1}{#2}}
\MT@end@catcodes

\MT@set@inh@list
\MT@map@clist@c\MT@features{\\KV@@sp@def\@tempa{##1}
\MT@ifempty\@tempa\relax{\MT@exp@one\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{\\KV@@sp@def\@tempa{##1}
\MT@ifempty\@tempa\relax{\MT@exp\MT@declare@char@inh
\csname MT@rbba@\@tempa\endcsname}{#1}{#2}}
\MT@end@catcodes

The lists cannot be given a name by the user.
Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations. We can reuse the key setup from the configuration lists (`\Set...`).

Now parse the third argument, the inheritance lists. We define the commands `\MT@inh@do`, 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`).

Only gather the inheriting characters here. Their codes will actually be set in `\MT@set@\<feature>\@codes`. 
14.3.7 Permutation

Calling \texttt{\textbackslash permute} will define commands for all permutations of the specified font attributes of the form \texttt{\textbackslash MT\{list type\}/\{encoding\}/\{family\}/\{series\}/\{shape\}/\{\*\}} to be the expansion of \texttt{\textbackslash MT\{list type\}/name}, i.e., the name of the currently defined list.

Size ranges are held in a separate macro called \texttt{\textbackslash MT\{list type\}/\{font axes\}/sizes}, which in turn contains the respective \texttt{\{list name\}}s attached to the ranges.

\begin{verbatim}
\documentclass{article}
\begin{document}
\footnotesize
\setlength{\fboxsep}{3pt}
\fbox{

In order to save some memory, we can ignore unused encodings (inside the document).

\end{verbatim}

\end{document}

}\end{verbatim}
Some sanity checks: an encoding must be specified (unless nothing else is).
\MT@ifstreq\@tempa{////}{\relax}{% You have to specify an encoding for the \@nameuse{MT@abbr@MT@permutelist} list \@nameuse{MT@MT@permutelist @name}'. Message Break
Ignoring it}%
\else
\MT@ifdefined@c@TF\MT@tempsize{ Add the list of ranges to the beginning of the current combination, after checking for conflicts.
\MT@ifdefined@n@T{MT@MT@permutelist @@tempaMT@extra@context @sises}{% Only one list can apply to a given combination. But we don't warn if the overridden list is to be loaded by the current one.
\MT@ifdefined@n@TF{MT@MT@permutelist @@tempaMT@extra@context @sises} \% \MT@ifstreq{\@nameuse{MT@MT@permutelist \@tempaMT@extra@context\@sizes}}{\relax}{% (debug) MT@dinfoonl{][}initialising: use list for font \@tempa, Message Break
(debug) sizes: \csname MT@MT@permutelist \@tempaMT@extra@context\@sizes\endcsname} \%
\else
\MT@ifdefined@c@OTF\MT@tempsize\% (debug) MT@dinfoonl{][}initialising: use list for font \@tempa
(debug) \csname MT@MT@permutelist \@tempaMT@extra@context\@sizes\endcsname} \%
\fi
\if\MT@extra\context\else Message Break font \@tempa\fi
\}\% (debug) MT@dinfoonl{][}initialising: use list for font \@tempa
\fi
\def\MT@permute@define#1{% \@tempcnta=\csname MT@cnt@#1\endcsname \relax \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta} {% \MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}} {\MT@let@nc{MT@temp#1}\@empty} }
\def\MT@permute@reset#1{% \@tempcnta=\@ne \MT@loop
\MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
\}Define the commands.
\MT@permute@define
\MT@permute@reset
Reset the commands.
\MT@check@rlist
For every new range item in \MT@tempsize, check whether it overlaps with ranges in the existing list.

\MT@check@rlist@
Define the current new range and ...

\MT@check@range
... recurse through the list of existing ranges.

\MT@check@range0
\@tempb and \@tempc are lower resp. upper bound of the new range, (#1) and (#2) those of the existing range. (#3) is the list name.

* Both items are simple sizes.

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

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

* Both items are ranges.
If we’ve already found a conflict with this item, we can skip the rest of the list.

14.4 Package options

14.4.1 Declaring the options

```latex
\ifMT@opt@expansion
\ifMT@opt@auto
\ifMT@opt@DVI
\MT@optwarn@admissible
Some warnings.
\def\MT@optwarn@admissible#1#2{%
\MT@warning@nl{`#1' is not an admissible value for option `#2'. Assuming `false'}%
\}
\MT@optwarn@nan
\def\MT@optwarn@nan#1#2{%
\MT@warning@nl{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@n@TF{MT@\@tempb @set@@\MT@val}{%\MT@xdef@n{MT@\@tempb @setname}{\MT@val}\
\MT@ifempty\MT@val\relax{\MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
\MT@warning@nl{The #1 set `\MT@val' is undeclared. MessageBreak Using set `\@nameuse{MT@\@tempb @setname}' instead}}%
\}
```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a \texttt{set name}.

```latex
\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{\MT@warning@nl{Value `#1' for option `#2' is not a number. Using default value of \number\@nameuse{MT@#2@default}}%
\}
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a \texttt{set name}.

```latex
\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{\MT@warning@nl{Value `#1' for option `#2' is not a number. Using default value of \number\@nameuse{MT@#2@default}}%
\}
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
\MT@warn@on{\MK@\@tempb @no{\MT@\@tempb}{\MT@opt@auto}{\MT@opt@DVI}}%
```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a \texttt{set name}.
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}}
}

\MT@map@clist@n{spacing,kerning,tracking}{
  \define@key{MT}{#1}[true]{
    \MT@map@clist@n{##1}{
      \KV@@sp@def\MT@val{####1}
      \MT@ifempty\MT@val\relax{\csname MT@#1true\endcsname}
      \MT@ifstreq\MT@val{true}\relax{
        \MT@ifstreq\MT@val{false}\relax{
          \MT@optwarn@admissible{##1}{#1}
          \edef\@tempa{false}
        }
      }
      \edef\@tempb{\csname MT@rbba@#1\endcsname}
      \MT@opt@def@set{#1}
    }
  }
}

\MT@def@bool@opt

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

\def\MT@def@bool@opt#1#2{\define@key{MT}{#1}[true]{\def\@tempa{##1}{\MT@ifstreq\@tempa{true}\relax{\MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{##1}{#1}{\def\@tempa{false}}}}}}

Boolean options that only set the switch.
The DVIoutput option will change \pdfoutput immediately to minimise the risk of confusing other packages.

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 options is also undocumented in the hope that we can always find out automatically whether it’s required. It also works with LuaTEX 0.30 or newer.
The `copyfonts' option does not work with xetex.

Use pdftex or luatex instead.

```
\ifxetex-def
\fi
\ifpdf-def\xetex-def\fi
```

final is the opposite to draft.

```
\csname if\@tempa\endcsname
\MT@draftfalse
\else\MT@drafttrue\fi
```

For verbose output, we redefine \MT@vinfo.

```
\define@key{MT}{verbose}[true]{
\let\MT@vinfo\MT@info@nl
\def\@tempa{#1}
\MT@ifstreq\@tempa{true}\relax{Take problems seriously.
\MT@ifstreq\@tempa{errors}{
\let\MT@warning \MT@warn@err
\let\MT@warning@nl\MT@warn@err
}{}
\let\MT@vinfo\@gobble
\MT@ifstreq\@tempa{silent}{
\let\MT@warning \MT@info
\let\MT@warning@nl\MT@info@nl
}{}
\MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}{}
}
```

```
\MT@requires@latex1{
\MT@map@clist@n{\MT@factor@default}\{stretch, shrink, step, letterspace}\{
\define@key{MT}{factor}[\MT@factor@default]{
\def\@tempa{#1}
\MT@ifint\@tempa
\MT@edef@n{MT@#1}{\@tempa}
\MT@optwarn@nan{##1}{#1}
}\}
```

```
\MT@requires@latex1{
\MT@factor@default
\define@key{MT}{factor}[\MT@factor@default]{
\def\@tempa{#1}
\MT@ifint\@tempa
\MT@edef@n{MT@#1}{\@tempa}
\MT@optwarn@nan{##1}{#1}
}\}
```

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

```
\MT@ifint\@tempa
\MT@edef@n{MT@#1}{\@tempa}
\MT@optwarn@nan{##1}{#1}
\}
\MT@ifint\@tempa
\MT@edef@n{MT@#1}{\@tempa}
\MT@optwarn@nan{##1}{#1}
\}
```

factor will define the protrusion factor only.

```
\define@key{MT}{factor}[\MT@factor@default]{
\def\@tempa{#1}
\MT@ifint\@tempa
```
Unit for protrusion codes.
\define@key{MT}{unit}{character}{% \def\@tempa{#1}% \MT@ifstreq\@tempa{character}\relax{% \MT@ifdimen\@tempa% \let\MT@pr@unit\@tempa% \MT@warning@nl{\@tempa is not a dimension.\MessageBreak Ignoring it and setting values relative to\MessageBreak character widths}% \}%}%

\subsection{Loading the definition file}
\MT@endinput Abort if no capable engine found.
\let\MT@endinput\relax
\ifx\MT@engine\relax\MT@warning@nl{You don't seem to be using pdftex, luatex or xetex.\MessageBreak \MT@MT only works with these engines.\MessageBreak I will quit now}\MT@clear@options\else\MT@requires@pdftex{\MT@expansiontrue\MT@autotrue}\fi
\MT@autotrue
\MT@endinput
\subsection{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's not likely that expanded fonts are available. (This grows more important as modern \TeX systems have switched to the pdftex engine even for DVI output, so that the user might not even be aware of the fact that she's running pdftex.)
\MT@protrusiontrue
\ifnum\pdfoutput<1\else\MT@requires@pdftex\MT@expansiontrue\MT@autotrue\fi
\MT@autotrue
\MT@endinput

Also, we only enable expansion by default if pdftex can expand the fonts automatically.
\MT@requires@pdftex\MT@autotrue
\MT@expansiontrue
\MT@endinput

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.
14.4.4 Hook for other packages

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

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using \@ifpackageloaded in the font package was not viable), and (2) checking \AtBeginDocument could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it’s simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren’t overwritten. Example:
\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\ifpackageloaded{microtype}
\MinionPro@MT@Hook
\else
\let\Microtype@Hook\MinionPro@MT@Hook
\if\undefined{Microtype@Hook}
\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.

14.4.5 Changing options later

\microtypesetup
\MT@define@optionX

Inside the preamble, \microtypesetup accepts the same options as the package (unless defersetup=false). In the document body, it accepts the options: protrusion, expansion, activate, tracking, spacing and kerning. Specifying font sets is not allowed.

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 MTO@rbba@level\endcsname
\MT@vinfo{Enabling #1 }
\{level \number\csname MTO@rbba@level\endcsname\}on@line%
}%

\MT@fstrreq{MTO@val}[false]{
\@tempcnta=\z@
\MT@vinfo{Disabling #1}on@line%
}%
\MT@fstrreq{MTO@val}[compatibility]{
\MT@checksetup[1]{
\@tempcnta=\one
\MT@let@nc{MTO@rbba@level}\one
\MT@vinfo{Setting #1 to level 1}on@line%
}%

}%
\MT@fstrreq{MTO@val}[nocompatibility]{
\MT@checksetup[1]{
\@tempcnta=\tw@
\MT@let@nc{MTO@rbba@level}\tw@
\MT@vinfo{Setting #1 to level 2}on@line%
}%

}%
\MT@error{Value \MT@val for key #1 not recognised}
\setboolean{MT@protrudechars}{true}
\setboolean{MT@adjustspacing}{true}
\ifnum\@tempcnta>\m@ne
\setboolean{MT@protrudechars}{true}
\fi
\ifnum\@tempcnta>\m@ne
\setboolean{MT@adjustspacing}{true}
\fi
\ifnum\@tempcnta<\z@ \setboolean{MT@protrudechars}{false} \setboolean{MT@adjustspacing}{false} \fi
\MT@checksetup{protrusion}
\MT@checksetup{adjustspacing}
\MT@checksetup{expansion}
\ifpdftex \let\pdfprotrudechars\protrudechars \let\pdfadjustspacing\adjustspacing \fi
\ifluatex \let\MT@protrudechars\pdfprotrudechars \let\MT@adjustspacing\pdfadjustspacing \fi
\ifxetex \let\XeTeXprotrudechars\pdfprotrudechars \let\XeTeXadjustspacing\pdfadjustspacing \fi
\define@key{MTX}{protrusion}[true]{\MT@warning{Ignoring protrusion setup}}
\define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
\define@key{MTX}{adjustspacing}[true]{\MT@warning{Ignoring adjust spacing setup}}
\ifusepackage{microtype} \else \MT@error{You cannot enable #1 if it was disabled in the package options}{Load microtype with #1 enabled.} \fi
\endinput
We cannot simply let \MT@tracking relax, since this may select the already letter-spaced font instance.

\MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero \else \let\MT@tracking\MT@tracking@ \fi}

\MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
\MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta \pdfappendkern\@tempcnta}

\MT@saved@setupfont Disable for older pdf\TeX versions and for \Xe\TeX and Lua\TeX.
\MT@define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
\MT@define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
\MT@define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
\MT@define@key{MTX}{activate}[true]{% 
\MT@ProcessOptionsWithKV{protrusion={#1}}% 
\MT@ProcessOptionsWithKV{expansion={#1}}%}
\MT@define@key{MTX}{disable}[]{% 
\MT@info{Inactivate \`MT\@MT package}% 
\let\MT@setupfont\relax}%
\MT@define@key{MTX}{enable}[]{% 
\MT@info{Reactivate \`MT\@MT package}% 
\let\MT@setupfont\MT@saved@setupfont}%
\MT@requires@latex2{\MT@map@clist@c\@classoptionslist{\CurrentOption{##1}%=}}
\MT@saved@setupfont Disable everything – may be used as a temporary work-around in case setting up fonts doesn’t work under certain circumstances, but only until that specific problem is fixed. This is undocumeneted, as it completely deprives us of the possibility to act – we’re blind and paralysed.

\MT@ProcessOptionsWithKV Parse options.
\MT@getkey For key=val in class options.
\def\MT@getkey#1=#2\@nil{#1}
\MT@ProcessOptionsWithKV{MT}
\relax
\relax
\MT@clear@options
\relax
\relax
\MT@addto@setup{\ifMT@draft
We disable most of what we've just defined in the 4419 lines above if we are running in draft mode.
\Warning\MessageBreak
Disabling all micro-typographic extensions.\MessageBreak
This might lead to different line and page breaks\%}
\let\MT@setupfont\relax
\renewcommand*\LoadMicrotypeFile[1]{\relax}
\renewcommand*\microtypesetup[1]{\relax}
\renewcommand*\microtypecontext[1]{\relax}
\renewcommand*\lsstyle{}\%}
\else
\MT@setup@PDF\MT@setup@copies
\relax
\relax
\MT@setup@PDF
\MT@setup@copies
\relax
\relax
\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
\relax
\relax
\MT@setup@PDF
pdflatex can create DVI output, too. However, both the DVI viewer and dvips need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

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

\MT@setup@copies Working on font copies?
\MT@setup@protrusion Protrusion.
\MT@setup@expansion For DVI output, the user must have explicitly passed the expansion option to the package. Under LuaTeX, expansion works quite differently: the glyphs will be positioned as if they were transformed, without actually being transformed. Since this could still be considered a viable option, we don't disable the feature completely, but issue a warning.

Set up the values for font expansion: if stretch has not been specified, we take the
If shrink has not been specified, it will inherit the value from stretch.

If step has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for microtype.pdf with step=1 compared to step=5). With older versions, we set it to min(stretch, shrink)/5, rounded off, minimum value 1.

\MT@auto
Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdfTeX). With LuaTeX, we just leave it empty, as there’s actually no difference – non-automatic font expansion doesn’t work anymore. In LuaTeX 1.0.6, the ‘autoexpand’ option seems to have been removed altogether and would trigger a warning.

We turn off automatic expansion if output mode is DVI.
If you have created expanded fonts instances, remove 'auto' from the package options. Otherwise, you have to switch off expansion completely.

Also, if pdfTeX is too old.

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.
\MT@check@step{stretch} \MT@check@step{shrink}
\MT@check@active@set{ex}
\MT@setup@tracking

Inside \showhyphens, font expansion should be disabled. (Since 2017/01/10, the \LaTeX{} format contains a different version for \TeX{}, but since expansion doesn’t work with \TeX{}, we don’t have to bother.) Since 2019/10/01, the command is robust.
\MT@setup@expansion

Tracking, spacing and kerning.
\MT@setup@tracking
Enable protrusion for compensation at the line edges.
\ifMT@protrusion\else\MT@protrudechars\fi
\else
\let\MT@tracking\relax
\MT@info\{No adjustment of tracking}\fi
</pdftex-def|luatex-def>
\MT@setup@spacing
\def\MT@setup@spacing{\ifMT@spacing
\edef\MT@active@features{\MT@active@features,sp}%
\pdfadjustinterwordglue\relax
\MT@info\{Adjustment of interword spacing enabled}%
The \ragged2e package sets interword spaces to a fixed value without glue. microtype's modifications can therefore have undesired effects. Therefore, we issue a warning.
\MT@with@package{T}{ragged2e}{%
\MT@warning\{You are using the `ragged2e' package. Adjustment of interword spacing may lead to undesired results when used with `ragged2e'.\MessageBreak In this case, disable the `spacing' option}%
\MT@check@active@set{sp}%
\else
\let\MT@spacing\relax
\MT@info\{No adjustment of interword spacing}%
\fi}
\MT@setup@spacing@check
Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue > 0. Why 1500? Because some packages redefine \frenchspacing.\footnote{Cf. the c.t.t. thread `frenchspacing with AMS packages and babel', started by Philipp Lehman on 16 August 2005, MID: dttbaj$rob$1@online.de}
\MT@setup@kerning
\def\MT@setup@kerning{\ifMT@kerning
\edef\MT@active@features{\MT@active@features,kn}%
\pdfprependkern\relax
\pdfappendkern\relax
\MT@info\{Adjustment of character kerning enabled}%
\MT@check@active@set{kn}%
\else
\let\MT@kerning\relax
\MT@info\{No adjustment of character kerning}%
\fi}
\MT@setup@kerning@check
If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error message. We also switch the features off for LuaTeX and XeTeX.

With pdfTeX, we issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn’t already, to compensate for the letterspacing kerns.
\MT@setup@noligatures\DisableLigatures is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

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

We patch the language switching commands to enable language-dependent setup.

Active characters can only be switched off if babel isn’t loaded after microtype.

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
\ifMT@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\fi\fi\fi
\ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi\fi
\ifMT@draft
\MT@addto@setup{\fi}
\MT@requires@latex1{
\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}}\relax
\MT@restore@catcodes
That was that.

In case babel was loaded before microtype:
\MT@set@babel@context\languagename
\if%
\MT@warning@nl{You did not load the babel package.\MessageBreak}
\MT@if@false\MT@warning@nl{The `babel' option won't have any effect}\%
\fi
\fi

Now we close the \fi from \ifMT@draft.
\MT@addto@setup{\fi}
Set up the current font, most likely the normal font. This has to come after all of
the setup (including anything from the preamble) has been dealt with.
\selectfont}
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.

\begin{verbatim}
\setcounter{section}{15}
\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.

\begin{verbatim}

\section{Font sets}

\subsection*{Config files}

Let's now write the font configuration files.
The default sets.

\DeclareMicrotypeSetDefault{protrusion}{alltext}
\DeclareMicrotypeSetDefault{expansion}{alltext-nott}
\DeclareMicrotypeSetDefault{spacing}{alltext-nott}
\DeclareMicrotypeSetDefault{kerning}{alltext}
\DeclareMicrotypeSetDefault{tracking}{smallcaps}

15.2 Font variants and aliases

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

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

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

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

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

4933 \DeclareMicrotypeAlias{New Computer Modern}{Latin Modern Roman}

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the TpX Gyre fonts Pagella and Termes (formerly: qfonts).

4934 \% -- Palatino
4935 \DeclareMicrotypeAlias{pxr}{ppr} \% pxfonts
4936 \DeclareMicrotypeAlias{qpl}{ppl} \% TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

The ‘FPL Neu’ fonts, a ‘re-implementation’ of Palatino.

4937 \DeclareMicrotypeAlias{fp9x}{pplx} \% FPL Neu
4938 \DeclareMicrotypeAlias{fp9j}{pplj} \% "

The newpx package, a replacement for pxfonts.

4939 \DeclareMicrotypeAlias{zpllf}{pplx} \% newpxtext
4940 \DeclareMicrotypeAlias{zpltosf}{pplj} \% "
4941 \DeclareMicrotypeAlias{zpltlf}{pplx} \% "
4942 \DeclareMicrotypeAlias{zpltosf}{pplj} \% "

The domitian package.

4943 \DeclareMicrotypeAlias{Domitian-TLF}{pplx} \% domitian
4944 \DeclareMicrotypeAlias{Domitian-TOsF}{pplj} \% "

The OpenType versions:

4945 \DeclareMicrotypeAlias{Domitian}{Palatino Linotype}
4946 \DeclareMicrotypeAlias{TeX Gyre Pagella}{Palatino Linotype}
4947 \DeclareMicrotypeAlias{Palatino LT Std}{Palatino Linotype}
4948 \DeclareMicrotypeAlias{Palatino}{Palatino Linotype}
4949 \DeclareMicrotypeAlias{Asana Math}{Palatino Linotype}
4950 \% -- Times New Roman
4951 \DeclareMicrotypeAlias{txr}{ptm} \% txfonts

The newtx package, a replacement for txfonts.

4952 \DeclareMicrotypeAlias{ntxlf}{ptmx} \% newtxtext
4953 \DeclareMicrotypeAlias{ntxosf}{ptmj} \% "
4954 \DeclareMicrotypeAlias{ntxtlf}{ptmx} \% "
4955 \DeclareMicrotypeAlias{ntxtosf}{ptmj} \% "

The tempora package.

4956 \DeclareMicrotypeAlias{Tempora-TLF}{ptmx} \% tempora
4957 \DeclareMicrotypeAlias{Tempora-TOsF}{ptmj} \% "
4958 \DeclareMicrotypeAlias{qtm}{ptm} \% TeX Gyre Termes (formerly: qfonts/QuasiTimes)

The step package.

4959 \DeclareMicrotypeAlias{STEP-TLF}{ptmx} \% step
4960 \DeclareMicrotypeAlias{STEP-TOsF}{ptmj} \% "

The stix and stix2 packages (the latter has departed a bit from being a Times clone, but still seems close enough).

4961 \DeclareMicrotypeAlias{stix}{ptm} \% stix
4962 \DeclareMicrotypeAlias{stix2}{ptm} \% stix2

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (TimesNewRomanMT, TimesNRCenturyMT), mtt (TimesSmallTextMT); pte (TimesEuropa); ptt (TimesTen); TimesEighteen; TimesModernMT.

MicroPress's Charter version (chmath).

4963 \% -- Charter
4964 \DeclareMicrotypeAlias{chr}{bch} \% CH Math

The XCharter package extends the Charter fonts.

4965 \DeclareMicrotypeAlias{XCharter-TLF}{bch} \% XCharter
4966 \DeclareMicrotypeAlias{XCharter-TOsF}{bch} \% "
The mathdesign package provides math fonts matching Bitstream Charter and URW Garamond.

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

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

\DeclareMicrotypeAlias{zgmx}{ugm} % garamondx
\DeclareMicrotypeAlias{zgmj}{ugm} % 
\DeclareMicrotypeAlias{zgmI}{ugm} % 
\DeclareMicrotypeAlias{zgmq}{ugm} % 

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

\DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch

The eulervm package virtually extends the Euler fonts.

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

Euro symbol fonts, to save some files.

\DeclareMicrotypeAlias{zpeus} {zpeu} % Adobe Euro sans -> serif
\DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
\DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif

15.3 Interaction with babel

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

\DeclareMicrotypeBabelHook {english,UKenglish,british,USenglish,american} {kerning=, spacing=nonfrench}
\DeclareMicrotypeBabelHook {french,francais,acadian,canadien} {kerning=french, spacing=} 
\DeclareMicrotypeBabelHook {turkish} {kerning=turkish, spacing=}

15.4 Note on admissible characters

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

\ : \textbackslash 
\{ : \textbraceright 
\} : \textbraceleft 
\^ : \textasciicircum 
\% : \%
\# : \#
Comma and equal sign must be guarded with braces (’, ‘{=’) to keep keyval happy.

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

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

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

With Xe\TeX or Lua\TeX, in contrast, it is advisable to use the proper Unicode characters, or the font-specific glyph names prefixed with ‘/’ (cf. section 16).

15.5 Character inheritance

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

Glyphs that should possibly inherit settings on one side only: 012 (‘fi’ ligature), 013 (‘ľ’), 014 (‘fi’), 015 (‘ff’), Ė, Ė, Œ, œ.

Candidates here: 028 (‘fi’), 029 (‘ľ’), 030 (‘fi’), 031 (‘ff’), 156 (‘IJ’ ligature, since \LaTeX 2005/12/01 accessible as \IJ), 188 (‘ji’, ‘ij’), Ė, Ė, Œ, œ.
G = {\u G},
g = {\u g},
I = {\`I,\'I,\^I,"I,\.I},
i = {\`i,\'i,\^i,"i,\i},
J = {\j},
L = {\L,\`L,\v L},
l = {\l,\`l,\v l},
N = {\`N,\~N,\v N},
n = {\~n,\.n,\n n},
o = {\o,\`o,\'o,\.o,\"o,\O,\H o},
R = {\`R,\v R},
r = {\`r,\v r},
S = {\s,\`s,\c s,\v s},
T = {\c T,\v T},
t = {\c t,\v t},
U = {\`U,\v U,\"U},
u = {\`u,\v u,\Y,\Y},
y = {\`y,\v y,\Y,\Y},
Z = {\c Z,\v Z},
z = {\c z,\v z,\v z}

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

% - = {127},
}

15.5.3 LY1

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

\DeclareCharacterInheritance
\{ encoding = LY1 }
\{
A = {\`A,\`A,\`A,\~A,\"A,\r A},
a = {\`a,\`a,\`a,\~a,\"a,\r a},
C = {\c C},
c = {\c c},
D = {\DH},
e = {\`e,\v e,\`e,\v e},
f = {011}, % ff
I = {\`I,\v I,\`I,\v I},
i = {\`i,\v i,\`i,\v i},
L = {\L},
l = {\L},
N = {\`N},
n = {\`n},
O = {\`O,\v O,\`O,\v O,\"O,\O,\H o},
o = {\`o,\v o,\`o,\v o,\"o,\O,\H o},
S = {\v S},
s = {\v s},
U = {\`u,\v u,\`u,\v u},
u = {\`u,\v u,\`u,\v u},
Y = {\`y,\v y,\Y,\Y},
y = {\`y,\v y,\Y,\Y},
Z = {\v Z},
z = {\v z}
\}
15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 (’f’), 012 (’fi’), 013 (’fl’), 014 (’ffi’), 015 (’ffl’), Æ, æ, Œ, œ.

\begin{verbatim}
\DeclareCharacterInheritance
  { encoding = OT4 }
  { A = \{k A\},
    a = \{k a\},
    C = \{\c 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
\end{verbatim}

15.5.5 QX

The Central European QX encoding.17 Ligatures: 009 (’fk’), 012 (’fi’), 013 (’fl’), 014 (’ffi’), 015 (’ffl’), Æ, æ, Œ, œ.

\begin{verbatim}
\DeclareCharacterInheritance
  { encoding = QX }
  { A = \{\`A,\a,\^A,\~A,\"A,\k A,\AA\},
    a = \{\`a,\a,\^a,\~a,\"a,\k a,\aa\},
    C = \{\c C,\c C\},
    c = \{\c c,\c c\},
    D = \{\DH\},
    E = \{\`E,\e,\^E,\"E,\k E\},
    e = \{\e,\e,\^e,\"e,\k e\},
    f = \{011\}, % ff
    I = \{\i,\i,\i,\i,\i,\i,\i\},
    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\},
  } \textquotedblleft = "FF
\end{verbatim}

The Romanian \textcommabelow accents are actually replacements for the \c variants, which had previously (and erroneously18) been included in QX encoding. They are still kept for backwards compatibility.

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

The EU1 (Xe\LaTeX), EU2 (Lua\LaTeX), and, since fontspec version 2.5, TU encodings are not well-defined in the sense that they don't contain a fixed number of glyphs, all of which must be present. OpenType fonts may contain thousands of glyphs, but we only define those that should be present in every font (basically T1). This inheritance list should be overridden by font-specific ones.
E = {\`E,\^E,\"E,\k E,\w E},
e = {\`e,\^e,\"e,\k e,\w e},
f = {/f,\_f}, % sometimes /f_f, sometimes /ff
G = {\u G},
g = {\u g},
I = {\`I,\^I,\"I,\.I},
i = {\`i,\^i,\"i,\.i},
j = {\j},
L = {\L,\`L,\v L},
l = {\l,\`l,\v l},
N = {\N,\~N,\v N},
n = {\n,\~n,\v n},
o = {\O,\`O,\^O,\~O,\"O,\H O},
o = {\o,\`o,\^o,\~o,\"o,\H o},
R = {\R,\v R},
r = {\r,\v r},
S = {\S,\c S,\v S}, % \SS
s = {\s,\c s,\v s},
T = {\c T,\v T},
t = {\c t,\v t},
U = {\U,\`U,\~U,\v U,\H U,\r U},
u = {\u,\`u,\~u,\v u,\H u,\r u},
Y = {\Y},
y = {\y},
Z = {\c Z,\Z,\v Z},
z = {\c z,\.z,\v z}
}

15.5.8 Euro symbols

Make Euro symbols settings simpler.

\DeclareCharacterInheritance
{ encoding = U,
family = {zpeu,zpeus,eurosans} }
{ E = 128 }

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

\DeclareCharacterInheritance
{ encoding = {OT1,U},
family = mvs }
{ 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm

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.
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).

```
\SetExpansion
[ name = default ]
{ encoding = {OT1,OT4,QX,T1,LY1} }
{
A = 500, a = 700,
\AE = 500, \ae = 700,
B = 700, b = 700,
C = 700, c = 700,
D = 500, d = 700,
E = 700, e = 700,
F = 700,
G = 500, g = 700,
H = 700, h = 700,
K = 700, k = 700,
M = 700, m = 700,
N = 700, n = 700,
O = 500, o = 700,
\OE = 500, \oe = 700,
P = 700, p = 700,
Q = 500, q = 700,
R = 700,
S = 700, s = 700,
U = 700, u = 700,
W = 700, w = 700,
Z = 700, z = 700,
2 = 700,
3 = 700,
6 = 700,
8 = 700,
9 = 700
}

Settings for Cyrillic T2A encoding.\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,
```
T5 encoding does not contain \AE, \ae, \OE and \oe.
15.8 Character protrusion

For future historians, Hàn Thế Thành's original settings (from protcode.tex, converted to microtype notation).

```
\SetProtrusion
[ name = thanh ]
[ encoding = OT1 ]
[ A = \{ 50, 50 \},
  F = \{ , 50 \},
  J = \{ 50, \},
  K = \{ , 50 \},
  L = \{ , 50 \},
  T = \{ 50, 50 \},
  V = \{ 50, 50 \},
  W = \{ 50, 50 \},
  X = \{ 50, 50 \},
  Y = \{ 50, 50 \},
  k = \{ , 50 \},
  r = \{ , 50 \},
  t = \{ , 50 \},
  v = \{ 50, 50 \},
  w = \{ 50, 50 \},
  x = \{ 50, 50 \},
  y = \{ 50, 50 \},
  . = \{ ., 700 \},
  , = \{ , 700 \},
  : = \{ ., 500 \},
  ; = \{ , 500 \},
  ! = \{ , 200 \},
  ? = \{ , 200 \},
  ( = \{ 50, \},
  ) = \{ , 50 \},
  = \{ ., 700 \},
  \textendash = \{ ., 300 \},
  \textemdash = \{ ., 200 \},
  \textquoteleft = \{ 700, \},
  \textquoteright = \{ ., 700 \},
  \textquotedblleft = \{ 500, \},
  \textquotedblright = \{ ., 500 \}]
```

15.8.1 Normal

The default settings always use the most moderate value.
* Bitstream Charter (NFSS code bch)

```plaintext
bch  [ name = bch-default ]
```

* Bitstream Letter Gothic (blg)

```plaintext
blg  [ name = blg-default ]
```

* Computer Modern Roman (cmr)

```plaintext
cmr  [ name = cmr-default ]
```

* Adobe Garamond (pad, padx, padj)

```plaintext
pad  [ name = pad-default ]
```

* Minion 20 (pmnx, pmnj)

```plaintext
pmn  [ name = pmnj-default ]
```

* Times (ptm, ptmx, ptmj)

```plaintext
ptm  [ name = ptm-default ]
```

* URW Garamond (ugm)

```plaintext
ugm  [ name = ugm-default ]
```

---

20 Contributed by Harald Harders and Karl Karlsson.
\begin{verbatim}
5401 \{(ugm) \OE = \{50,50\},
5402 \{blg\} P = \{ .100\},
5403 \{ugm\} P = \{ .50\},
5404 \{bch|pad|pmn\} Q = \{50,70\},
5405 \{ugm\} Q = \{50,50\},
5406 \{bch\} R = \{ .50\},
5407 \{ugm\} R = \{ .70\},
5408 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} T = \{50,50\},
5409 \{blg\} T = \{100,100\},
5410 \{ugm\} T = \{70,70\},
5411 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} V = \{50,50\},
5412 \{blg\|ugm\} V = \{70,70\},
5413 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} W = \{50,50\},
5414 \{ugm\} W = \{70,70\},
5415 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} X = \{50,50\},
5416 \{ugm\} X = \{50,70\},
5417 \{m-t\|bch|cmr|pad|pmn|ppl\} Y = \{50,50\},
5418 \{blg\|ptom|ugm\} Y = \{80,80\},
5419 \{ugm\} Z = \{50,50\},
5420 \{blg\} T = \{150,100\},
5421 \{blg\} t = \{150,150\},
5422 \{blg\} j = \{100,100\},
5423 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} k = \{ .50\},
5424 \{ugm\} k = \{ .70\},
5425 \{blg\} l = \{150,150\},
5426 \{pmn\} l = \{ .50\},
5427 \{pad|ppl\} p = \{50,50\},
5428 \{ugm\} p = \{ .50\},
5429 \{pad|ppl\} q = \{50\},
5430 \{blg\} r = \{ .50\},
5431 \{blg\} r = \{100,80\},
5432 \{cmr|pad|pmn\} t = \{ .70\},
5433 \{bch\} t = \{ .50\},
5434 \{blg\} t = \{150,80\},
5435 \{ugm\} t = \{ .100\},
5436 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} v = \{50,50\},
5437 \{blg\} v = \{100,100\},
5438 \{ugm\} v = \{50,70\},
5439 \{m-t\|bch|cmr|pad|pmn|ppl|ptm\} w = \{50,50\},
5440 \{ugm\} w = \{50,70\},
5441 \{blg\} x = \{50,50\},
5442 \{blg\} x = \{100,100\},
5443 \{m-t\|bch|pad|pmn\} y = \{ .50\},
5444 \{blg\} y = \{50,100\},
5445 \{cmr|ppl|ptom\} y = \{50,70\},
5446 \{ugm\} y = \{ .70\},
5447 \{cmr\} 0 = \{ .50\},
5448 \{m-t\} 1 = \{50,50\},
5449 \{bch|blg|pad|ptom|ugm\} 1 = \{150,150\},
5450 \{cmr\} 1 = \{100,200\},
5451 \{pmn\} 1 = \{ .50\},
5452 \{ppl\} 1 = \{100,100\},
5453 \{bch|cmr|pad|ugm\} 2 = \{50,50\},
5454 \{blg\} 2 = \{100\},
5455 \{bch|pmn\} 3 = \{50\},
5456 \{cmr|pad|ugm\} 3 = \{50,50\},
5457 \{blg\} 3 = \{100\},
5458 \{m-t|ptom\} 4 = \{50,50\},
5459 \{bch\} 4 = \{100,50\},
5460 \{blg\} 4 = \{100\},
5461 \{cmr|ugm\} 4 = \{70,70\},
5462 \{pmn\} 4 = \{50\},
5463 \{ptom\} 4 = \{70\},
5464 \{cmr\} 5 = \{ .50\},
5465 \{pad\} 5 = \{50,50\},
\end{verbatim}

\textbf{CONFIGURATION FILES: Character protrusion 155}
CONFIGURATION FILES: Character protrusion

5466 \( \text{bch} \) 6 = \{50, 1\},
5467 \( \text{cmr} \) 6 = \{ .50\},
5468 \( \text{pad} \) 6 = \{50,50\},
5469 \( \text{m-t} \) 7 = \{50,50\},
5470 \( \text{bch} | \text{pad} | \text{pmn} | \text{ugm} \) 7 = \{50,80\},
5471 \( \text{big} \) 7 = \{100,100\},
5472 \( \text{cmr} | \text{ptm} \) 7 = \{50,100\},
5473 \( \text{ppl} \) 7 = \{ .50\},
5474 \( \text{cmr} \) 8 = \{ .50\},
5475 \( \text{bch} | \text{pad} \) 9 = \{50,50\},
5476 \( \text{cmr} \) 9 = \{ .50\},
5477 \( \text{m-t} | \text{cmr} | \text{pad} | \text{pmn} | \text{ppl} | \text{ptm} | \text{ugm} \) \cdot = \{.700\},
5478 \( \text{bch} \) = \{ .600\},
5479 \( \text{big} \) = \{400,500\},
5480 \( \{ \text{big} \} \) \{.\} = \{.500\},
5481 \( \{ \text{big} \} \{.\} = \{300,400\},
5482 \( \text{m-t} | \text{cmr} | \text{pad} | \text{pmn} | \text{ppl} | \text{ptm} | \text{ugm} \) : = \{.500\},
5483 \( \text{bch} \) = \{400\},
5484 \( \{ \text{big} \} \) = \{300,400\},
5485 \( \text{m-t} | \text{bch} | \text{pad} | \text{pmn} | \text{ptm} \) ; = \{.300\},
5486 \( \{ \text{big} \} \) = \{(200,300)\},
5487 \( \text{cmr} | \text{ppl} \) ; = \{.500\},
5488 \( \text{ugm} \) = \{(400)\},
5489 \( \{ \text{big} \} \) \{\} = \{(100)\},
5490 \( \{ \text{big} \} \{\} = \{(200,200)\},
5491 \( \text{m-t} | \text{pad} | \text{pmn} | \text{ptm} \) ? = \{100\},
5492 \( \text{bch} | \text{cmr} | \text{ppl} | \text{ugm} \) ? = \{(200)\},
5493 \( \{ \text{big} \} \) \{\} = \{(150,150)\},
5494 \( \text{pmn} \) \& = \{(300,300)\},
5495 \( \text{m-t} | \text{bch} | \text{cmr} | \text{pad} | \text{pmn} | \text{ppl} \) \& = \{(50,50)\},
5496 \( \text{ptm} \) \& = \{(100,100)\},
5497 \( \text{m-t} | \text{bch} | \text{big} | \text{cmr} | \text{pad} | \text{pmn} | \text{ppl} | \text{ptm} \) = \{(200,250)\},
5498 \( \text{ugm} \) = \{(300,350)\},
5499 \( \text{pad} | \text{ppl} | \text{ptm} \) \& = \{(50,100)\},
5500 \( \text{ugm} \) \& = \{(100)\},
5501 \( \text{m-t} | \text{cmr} | \text{pad} | \text{pmn} \) \& = \{(50,50)\},
5502 \( \text{bch} \) \& = \{(50)\},
5503 \( \text{ppl} | \text{ptm} \) \& = \{(100,100)\},
5504 \( \text{ugm} \) \& = \{(50,100)\},
5505 \( \{ \text{big} \} \) \& = \{(100,100)\},
5506 \( \text{m-t} | \text{ppl} | \text{ptm} | \text{ugm} \) \& = \{(200,200)\},
5507 \( \text{bch} | \text{pmn} \) \& = \{(200,300)\},
5508 \( \{ \text{big} \} \) \& = \{(150,200)\},
5509 \( \text{cmr} | \text{pad} \) \& = \{(300,300)\},
5510 \( \text{m-t} | \text{cmr} | \text{ppl} | \text{ptm} \) \& = \{(250,250)\},
5511 \( \text{bch} \) \& = \{(150,250)\},
5512 \( \text{pad} \) \& = \{(300,300)\},
5513 \( \{ \text{big} \} | \text{pmn} \) \& = \{(150,200)\},
5514 \( \text{ugm} \) \& = \{(250,300)\},
5515 \( \{ \text{big} \} | \text{ugm} \) \& = \{(200,200)\},
5516 \( \text{m-t} | \text{pad} | \text{pmn} | \text{ptm} \) \& = \{(100,200)\},
5517 \( \text{bch} | \text{ugm} \) \& = \{(200,400)\},
5518 \( \text{cmr} | \text{big} \) \& = \{(300,500)\},
5519 \( \{ \text{big} \} \) \& = \{(100,200)\},
5520 \( \{ \text{big} \} \) \& = \{(300,400)\},
5521 \( \{ \text{big} \} \) \& = \{(300,500)\},
Why settings for left and right quotes? Because in some languages they might be used like that (see the csquotes package for examples).

Greek uppercase letters are in OT1 encoding only.

Remaining slots can be found in the source file.

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For XeTeX (EU1) and LuaTeX (EU2) we simply use the T1 list as default (for now).
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 \texttt{lmodern} fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

\begin{verbatim}
\SetsProtrusion{ encoding = {T1,LY1}, family = lmr }
{ \textquotedblleft = {300,400}, \textquotedblright = {300,400} }
\end{verbatim}

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).\footnote{Contributed by Karl Karlsson.}

\begin{verbatim}
\SetsProtrusion{ encoding = T2A, family = cmr }
{ \CYRA = {50,50}, \CYRG = {50,50}, \CYRK = {50,50}, \CYRT = {50,50}, \CYRU = {50,50} }
\setspecials{ \cyrs = {50,50} }
\end{verbatim}
Settings for the QX encoding (generic and Times).\textsuperscript{22} It also includes some glyphs otherwise in TS1.

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 [ name = pmnx-OT1, load = pmnj-default ]
{ encoding = OT1, family = pmnx }

\SetProtrusion [ name = pmnx-T1, load = pmnj-T1 ]
{ encoding = {T1,LY1}, family = pmnx }

\SetProtrusion [ name = pmnx-T2A, load = pmnj-T2A ]
{ encoding = {T2A}, family = pmnx }

\SetProtrusion [ name = ptm-LY1, load = ptm-T1 ]
{ encoding = LY1, family = {ptm,ptmx,ptmj} }

\SetProtrusion [ -- = {100,100},
\texttrademark = {100,100}, \textregistered = {100,100}, \textcopyright = {100,100}, \textdegree = {300,300}, \textminus = {200,200}, \textellipsis = {150,200},
\textless = {200,100}, \textgreater = {100,200}]

\SetProtrusion [ name = pmnx-T2A, load = pmnj-T2A ]
{ encoding = {T2A}, family = pmnx }

\SetProtrusion [ name = ptm-LY1, load = ptm-T1 ]
{ encoding = LY1, family = {ptm,ptmx,ptmj} }

\SetProtrusion [ -- = {100,100},
\texttrademark = {100,100}, \textregistered = {100,100}, \textcopyright = {100,100}, \textdegree = {300,300}, \textminus = {200,200}, \textellipsis = {150,200},
\textless = {200,100}, \textgreater = {100,200}]

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.
15.8.2 Italics

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

The italic glyphs of Computer Modern Roman feature a lot of side bearing, therefore almost all of them have to protrude.23

Settings contributed by Hendrik Vogt.
CONFIGURATION FILES: Character protrusion

\$ = \{20, -30\},  \;
\rangle = \{100, -55\},  \;
S = \{90, -65\},  \;
R = \{20, \},  \;
Q = \{70, 50\},  \;
P = \{82, -50\},  \;
O = \{70, 50\},  \;
N = \{50, \},  \;
M = \{50, \},  \;
L = \{50, \},  \;
K = \{50, \},  \;
J = \{50, \},  \;
I = \{50, \},  \;
G = \{50, \},  \;
F = \{50, \},  \;
E = \{50, \},  \;
D = \{50, \},  \;
C = \{50, \},  \;
B = \{50, \},  \;
A = \{50, \}.
\[ T = (220, -85), \]
\[ U = (230, -55), \]
\[ V = (260, -60), \]
\[ W = (185, -55), \]
\[ X = (70, -30), \]
\[ Y = (250, -60), \]
\[ Z = (90, -60), \]
\[ a = (150, -10), \]
\[ b = (170, ), \]
\[ c = (173, -10), \]
\[ d = (150, -55), \]
\[ e = (180, ), \]
\[ f = (50, 50), \]
\[ g = (155, ), \]
\[ h = (100, ), \]
\[ i = (210, ), \]
\[ j = (70, -30), \]
\[ k = (110, -50), \]
\[ l = (240, -110), \]
\[ m = (80, ), \]
\[ n = (115, ), \]
\[ o = (50, 50), \]
\[ p = (155, ), \]
\[ q = (150, -10), \]
\[ r = (155, -40), \]
\[ s = (110, -50), \]
\[ t = (230, -10), \]
\[ u = (120, ), \]
\[ v = (140, -25), \]
\[ w = (50, 50), \]
\[ x = (65, -40), \]
\[ y = (130, -20), \]
\[ z = (110, -80), \]
\[ 0 = (170, -85), \]
\[ 1 = (150, 100), \]
\[ 2 = (230, 110), \]
\[ 3 = (150, ), \]
\[ 4 = (50, ), \]

CONFIGURATION FILES: Character protrusion
CONFIGURATION FILES: Character protrusion

```
6000 (ppl) 1 = \{100, \},
6001 (ugm) 1 = \{150,150\},
6002 (cmr) 2 = \{130, -70\},
6003 (pad|ppl|ptm) 2 = \{50, \},
6004 (pmn) 2 = \{150,150\},
6005 (bch) 3 = \{140, -70\},
6006 (cmr) 3 = \{-100, \},
6007 (ppl) 3 = \{100,50\},
6008 (cmr) 4 = \{130,80\},
6009 (pad|ppl) 4 = \{50, \},
6010 (cmr) 5 = \{160, \},
6011 (ptm) 5 = \{50, \},
6012 (bch) 6 = \{50, \},
6013 (cmr) 6 = \{175,-30\},
6014 (ppl|ptm) 6 = \{100, \},
6015 (cmr) 7 = \{250,-150\},
6016 (pmn) 7 = \{20, \},
6017 (ppl) 7 = \{50, \},
6018 (cmr) 8 = \{130, -40\},
6019 (pad) 8 = \{150, \},
6020 (ppl|ptm) 8 = \{50, \},
6021 (cmr) 9 = \{155,-80\},
6022 (m-t|cmr|pad|pmn|ppl) = \{500\},
6023 (blg) = \{400,600\},
6024 (m-t|pad|pmn|ppl) = \{700\},
6025 (blg) = \{300,500\},
6026 (m-t|pad|pmn) = \{500\},
6027 (pmn) = \{200\},
6028 (ppl) = \{500\},
6029 (cmr) = \{130\},
6030 (cmr) = \{155\},
6031 (cmr) = \{180\},
6032 (cmr) = \{450\},
6033 (ppl) = \{100\},
6034 (cmr) = \{130\},
6035 (cmr) = \{155\},
6036 (cmr) = \{180\},
6037 (cmr) = \{180\},
6038 (cmr) = \{180\},
6039 (cmr) = \{180\},
6040 (cmr) = \{180\},
6041 (cmr) = \{180\},
6042 (cmr) = \{180\},
6043 (cmr) = \{180\},
6044 (cmr) = \{180\},
6045 (cmr) = \{180\},
6046 (cmr) = \{180\},
6047 (cmr) = \{180\},
6048 (cmr) = \{180\},
6049 (cmr) = \{180\},
6050 (cmr) = \{180\},
6051 (cmr) = \{180\},
6052 (cmr) = \{180\},
6053 (cmr) = \{180\},
6054 (cmr) = \{180\},
6055 (cmr) = \{180\},
6056 (cmr) = \{180\},
6057 (cmr) = \{180\},
6058 (cmr) = \{180\},
6059 (cmr) = \{180\},
6060 (cmr) = \{180\},
6061 (cmr) = \{180\},
6062 (cmr) = \{180\},
6063 (cmr) = \{180\},
6064 (cmr) = \{180\},
```
\texttt{configuration files: Character protrusion}

\begin{verbatim}
6065 \texttt{(ptm) \( \Theta = \{150,150\}, \)}
6066 \texttt{(m-t|bch|ugm) \(- = \{150,150\}, \)}
6067 \texttt{(cmr|pad|pmn|ppl|ptm) \(- = \{200,150\}, \)}
6068 \texttt{(ugm) \(+ = \{200,200\}, \)}
6069 \texttt{(m-t|bch|pad|pmn|ppl|ptm|ugm) \{ = \{200,\}, \} = \{200,\}, \)}
6070 \texttt{(cmr) \{ = \{300,\}, \} = \{300,\}, \)}
6071 \texttt{(m-t|pad|ppl|ptm|ugm) / = \{100,200\}, \)}
6072 \texttt{(cmr) / = \{100,100\}, \)}
6073 \texttt{(bch) / = \{150,\}, \)}
6074 \texttt{(pmn) / = \{100,150\}, \)}
6075 \texttt{(m-t) - = \{300,300\}, \)}
6076 \texttt{(bch|pad) - = \{300,400\}, \)}
6077 \texttt{(pmn) - = \{200,300\}, \)}
6078 \texttt{(cmr) - = \{500,300\}, \)}
6079 \texttt{(pmn) - = \{300,500\}, \)}
6080 \texttt{(ptm) - = \{500,500\}, \)}
6081 \texttt{(ugm) - = \{400,700\}, \)}
6082 \texttt{(big) \( = \{0,300\}, \)}
6083 \texttt{(m-t|pmn) \texttt{\textbackslash{textemdash} = \{200,200\}, \texttt{\textemdash} = \{150,150\}, \})
6084 \texttt{(bch) \texttt{\textbackslash{textemdash} = \{200,300\}, \texttt{\textemdash} = \{150,200\}, \})
6085 \texttt{(cmr) \texttt{\textemdash} = \{500,300\}, \texttt{\textemdash} = \{400,170\}, \})
6086 \texttt{(pad|ppl|ptm|ugm) \texttt{\textbackslash{textemdash} = \{300,300\}, \texttt{\textemdash} = \{200,200\}, \})
6087 \texttt{(m-t|bch|pmn|ugm) \texttt{\textbackslash{textemdash} = \{200,200\}, \texttt{\textemdash} = \{150,150\}, \})
6088 \texttt{(cmr) \texttt{\textemdash} = \{400,200\}, \texttt{\textemdash} = \{400,200\}, \})
6089 \texttt{(big) \texttt{\textendash} = \{400,400\}, \texttt{\textemdash} = \{400,400\}, \})
6090 \texttt{(cmr) \texttt{\textendash} = \{800,200\}, \texttt{\textemdash} = \{800,-20\}, \})
6091 \texttt{(pad) \texttt{\textendash} = \{800,200\}, \texttt{\textemdash} = \{800,200\}, \})
6092 \texttt{(pmn) \texttt{\textendash} = \{700,400\}, \texttt{\textemdash} = \{700,400\}, \})
6093 \texttt{(ptm) \texttt{\textendash} = \{800,500\}, \texttt{\textemdash} = \{800,500\}, \})
6094 \texttt{(ugm) \texttt{\textendash} = \{600,200\}, \texttt{\textemdash} = \{600,200\}, \})
6100 \}
6101 \)
6102 \{ \texttt{cmr|pmn} \}
6103 \texttt{\textbackslash{SetProtrusion}}
6104 \texttt{(cmr) \{ name = cmr-it-OT1, \}}
6105 \texttt{(pmn) \{ name = pmn-it-OT1, \}}
6106 \texttt{(cmr) \{ load = cmr-it \}}
6107 \texttt{(pmn) \{ load = pmn-it \}}
6108 \texttt{(cmr) \{ encoding = \{OT1,074\}, \}}
6109 \texttt{(pmn) \{ encoding = \{OT1, \}}
6110 \texttt{(cmr) \{ family = cmr, \}}
6111 \texttt{(pmn) \{ family = pmn, \}}
6112 \texttt{(cmr) \{ shape = \{it,sl\} \}}
6113 \texttt{(pmn) \{ shape = \{it,sl\} \}}
6114 \}
6115 \texttt{(cmr) \\AE = \{100, \}}
6116 \texttt{(pmn) \\AE = \{100, \}}
6117 \texttt{(cmr) \\OE = \{100, \}}
6118 \texttt{(pmn) \\OE = \{50, \}}
6119 \{ \texttt{cmr} \}
6120 \texttt{\"00 = \{200,150\}, \% \Gamma Gamma \}}
6121 \texttt{\"01 = \{150,100\}, \% \Delta Delta \}}
6122 \texttt{\"02 = \{150,50\}, \% \Theta Theta \}}
6123 \texttt{\"03 = \{150,50\}, \% \Lambda Lambda \}}
6124 \texttt{\"04 = \{100,100\}, \% \Xi Xi \}}
6125 \texttt{\"05 = \{100,100\}, \% \Pi Pi \}}
6126 \texttt{\"06 = \{100,50\}, \% \Sigma Sigma \}}
6127 \texttt{\"07 = \{200,150\}, \% \Upsilon Upsilon \}}
6128 \texttt{\"08 = \{150,50\}, \% \Phi Phi \}}
6129 \texttt{\"09 = \{150,100\}, \% \Psi Psi \}}
"OA = \{ 50, 50 \} % \Omega

\[( \text{cmr} \) \]

```
\setprotrusion\( \langle m-t \rangle \) [ name = T1-it-default, \\
( bch \) [ name = bch-it-T1, \\
( blg \) [ name = blg-it-T1, \\
( cmr \) [ name = cmr-it-T1, \\
( pmn \) [ name = pmnj-it-T1, \\
( ppl \) [ name = ppl-it-T1, \\
( ptm \) [ name = ptm-it-T1, \\
( ugm \) [ name = ugm-it-T1, \\
\m-t \) load = OT1-it ] \\
\bch \) load = bch-it ] \\
\blg \) load = blg-T1 ] \\
\cmr \) load = cmr-it ] \\
\pmn \) load = pmnj-it ] \\
\pad \) load = pad-it ] \\
\ppl \) load = ppl-it ] \\
\ptm \) load = ptm-it ] \\
\ugm \) load = ugm-it ]

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\bch \} family = bch, \\
\blg \} family = blg, \\
\cmr \} family = cmr, \\
\pmn \} family = pmnj, \\
\pad \} family = \{pad,padx,padj\}, \\
\ppl \} family = \{ppl,pplx,pplj\}, \\
\ptm \} family = \{ptm,ptmx,ptmj\}, \\
\ugm \} family = ugm

\m-t|bch|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it

\m-t|bch|cmr|pad|pmn|ppl|ptm \{ encoding = \{T1,LY1\}, \\
\blg|ptm|ugm \{ encoding = T1, \\
\m-t|bch|cmr|pad|pmn|ppl|ptm \{ shape = \{it,sl\} } \\
\blg|ptm|ugm \} shape = it
CONFIGURATION FILES: Character protrusion

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}

\textexclamdown = \{100, \}, \textquestiondown = \{200, \},
\textbraceleft = \{200,100\}, \textbraceright = \{200,200\},
\textless = \{100, \}, \textgreater = \{100, \},
\textvisiblespace = \{100,100\}
6260 (cmr)  \textcyrs = (100, ),
6261 (pmn)  \textcyrs = (50, ),
6262 (cmr)  \textcyrt = (100, ),
6263 (pmn)  \textcyrt = (70, ),
6264 (cmr)  \textcyru = (100, ),
6265 (pmn)  \textcyru = (50, ),
6266 (cmr)  \textcyrf = (100, ),
6267 (cmr)  \textcyrh = (50, ),
6268 (cmr)  \textcyrc = (50, ),
6269 (cmr)  \textcyrch = (100, ),
6270 (cmr)  \textcyrsh = (50, ),
6271 (cmr)  \textcyrsht = (50, ),
6272 (cmr)  \textcyrhdos = (100, ),
6273 (cmr)  \textcyrey = (50, ),
6274 (cmr)  \textcyrftsln = (50, ),
6275 (cmr)  \textcyrerev = (50, ),
6276 (cmr)  \textcyru = (50, ),
6277 (cmr)  \textcyra = (50, ),
6278 (pmn)  \textcyra = (20, ),
6279 (pmn)  \textcyrr = (50, ),
6280 (m-t|pmn)  _ = (0, 100),
6281 (cmr)  _ = (100,200),
6282 (pmn)  \textless = (300,100), \textgreater = (200,100)
6283 (cmr)  \textless = (300,100), \textgreater = (200,100)
6284 (m-t) \textbackslash = (100, ), \quotedblbase = (150,500),
6285 (cmr) \textbackslash = (100, ), \quotedblbase = (150,500),
6286 (pmn) \textbackslash = (100, ), \quotedblbase = (150,500),
6287 (m-t) \guillemotleft = (300,300), \guillemotright = (300,300),
6288 (cmr) \guillemotleft = (400,100), \guillemotright = (400,100),
6289 (pmn) \guillemotleft = (200,300), \guillemotright = (200,300),
6290 (m-t) \textbraceleft = (200,100), \textbraceright = (200,100),
6291 (cmr) \textbraceleft = (400,100), \textbraceright = (400,100),
6292 (pmn) \textbraceleft = (200,100), \textbraceright = (200,100),
6293 (cmr) \textquotedblleft = (500,300),
6294 (cmr) \textless = (300,100), \textgreater = (200,100)
6295 (pmn) \textless = (100, ), \textgreater = (100, )
6296   }
6297  
6298 (/m-t\cmr|pmn)
6299 \end{m-t}\ptm
6300 } \setprotrusion
6301 (m-t) [ name = QX-it-default,
6302 (ptm) [ name = ptm-it-QX,
6303 (m-t) \textload = OT1-it ]
6304 (ptm) \textload = ptm-it ]
6305 ( encoding = (QX),
6306 (ptm) \family = {ptm,ptmxx,ptmj}),
6307 \shape = {it,sl} ]
6308 (}
6309 (ptm) \text9 = (50, 50), \% fk
6310 (\textunderscore = (100,100),
6311 (ptm) \textunderscore = (100,100),
6312 (ptm) \textunderscore = (100,150),
6313 \textbackslash = (100,200),
6314 \Steelbase = (300,400),
6315 (m-t) \guillemotleft = (300,300), \guillemotright = (300,300),
6316 (ptm) \guillemotleft = (200,400), \guillemotright = (200,400),
6317 \textland = (200,100), \textquestiondown = (200,100),
6318 \textbraceleft = (200,100), \textbraceright = (200,100),
6319 \textless = (100,100), \textgreater = (100,100),
6320 \textmu = (200,200), \textdegree = (300,150),
6321 (m-t) \copyright = (100,100), \textregistered = (100,100)
6322 (ptm) \textregistered = (100,150), \copyright = (100,150),
6323 (ptm) \textdelta = (70, ), \textdelta = (50, ),
6324 (ptm) \textsf = (50, 80), \textmu = (80, 80),
Slanted is very similar to italic.

```latex
\SetProtrusion
\[ name = cmr-sl, \]
\load = cmr-it-OT1 \]
\{ encoding = {OT1,OT4}, \]
\family = cmr, \]
\shape = sl \}

\SetProtrusion
\[ name = cmr-sl-T1, \]
\load = cmr-it-T1 \]
\{ encoding = {T1,LY1}, \]
\family = cmr, \]
\shape = sl \}
```
\SetProtrusion
\[\text{name} = \text{cmr-sl-T2A,}\]
\[\text{load} = \text{cmr-it-T2A}\]
\[\text{encoding} = \text{T2A,}\]
\[\text{family} = \text{cmr,}\]
\[\text{shape} = \text{sl}\]

\SetProtrusion
\[\text{name} = \text{cmr-sl-T5,}\]
\[\text{load} = \text{cmr-it-T5}\]
\[\text{encoding} = \text{T5,}\]
\[\text{family} = \text{cmr,}\]
\[\text{shape} = \text{sl}\]

\SetProtrusion
\[\text{name} = \text{lmr-it-T1,}\]
\[\text{load} = \text{cmr-it-T1}\]
\[\text{encoding} = \{\text{T1},\text{LY1}\},\]
\[\text{family} = \text{lmr,}\]
\[\text{shape} = \{\text{it},\text{sl}\}\]

\textquotingleft\textquotedblleft = \{200,\}, \textquotingright\textquotedblright = \{200,\},
\textquotingbase\textquotedblslant = \{400,\}, \textquotingbase\textquotedblbase = \{500\}

Oldstyle numerals are slightly different.

\SetProtrusion
\[\text{name} = \text{cmr(oldstyle)-it,}\]
\[\text{load} = \text{cmr-it-T1}\]
\[\text{encoding} = \{\text{T1},\text{LY1}\},\]
\[\text{family} = \{\text{hfor},\text{cmor}\},\]
\[\text{shape} = \{\text{it},\text{sl}\}\]

1 = \{250, 50\},
2 = \{150, -100\},
3 = \{100, -50\},
4 = \{150, 150\},
6 = \{200,\},
7 = \{200, 50\},
8 = \{150, -50\},
9 = \{100, 50\}
CONFIGURATION FILES: Character protrusion

```latex
\SetProtrusion
\[\text{name} = \text{pmnx-it-T1},\]
\[\text{load} = \text{pmnj-it}\]
\{\text{encoding} = \{\text{T1}, \text{LY1}\},\]
\[\text{family} = \text{pmnx},\]
\[\text{shape} = \{\text{it}, \text{sl}\}\}
\{\text{1} = \{100, 150\}\}
\SetProtrusion
\[\text{name} = \text{pmnx-it-T2A},\]
\[\text{load} = \text{pmnj-it-T2A}\]
\{\text{encoding} = \{\text{T2A}\},\]
\[\text{family} = \text{pmnx},\]
\[\text{shape} = \{\text{it}, \text{sl}\}\}
\{\text{1} = \{100, 150\}\}
\SetProtrusion
\[\text{name} = \text{ptm-it-LY1},\]
\[\text{load} = \text{ptm-it-T1}\]
\{\text{encoding} = \{\text{LY1}\},\]
\[\text{family} = \{\text{ptm, pmnx, ptmj}\},\]
\[\text{shape} = \{\text{it}, \text{sl}\}\}
\{\text{1} = \{100, 150\}\}
\SetProtrusion
\{\text{pmn}\}
\{\text{ptm}\}
\{\text{ptm}\}
15.8.3 Small caps

Small caps should inherit the values from their big brothers. Since values are relative to character width, we don’t need to adjust them any further (but we have to reset some characters).

```plaintext
{+t(big|ugm)}
\SetProtrusion
{m-t} [ name = OT1-sc,
{bch} [ name = bch-sc,
{cmr} [ name = cmr-sc-OT1,
{pad} [ name = pad-sc,
{pmn} [ name = pmnj-sc,
{ppl} [ name = ppl-sc,
{ptm} [ name = ptm-sc,
{m-t} load = default ]
{bch} load = bch-default ]
{cmr} load = cmr-OT1 ]
{pad} load = pad-default ]
{pmn} load = pmnj-default ]
{ppl} load = ppl-default ]
{ptm} load = ptm-default ]
{m-t|bch|pad|pmn} ( encoding = OT1,
{cmr|ppl|ptm} ( encoding = {OT1,OT4},
{bch} family = bch,
{cmr} family = cmr,
{pad} family = {pad,padx,padjj},
{pmn} family = pmnj,
{ppl} family = {ppl,pplx,pplj},
{ptm} family = {ptm,ptmx,ptmj},
shape = sc )

{a = {50,50},
{bch|pad|ppl|ptm} \ae = {50, },
{bch|pad|pmn} c = {50, },
{bch|pad|pmn} d = { ,50},
{m-t|bch|cmr|pad|pmn|ptm} f = { ,50},
{bch|pad|pmn} g = {50, },
{m-t|cmr|pad|pmn|ppl|ptm} j = {50, },
{bch} j = {100, },
{m-t|bch|cmr|pad|pmn|ppl} l = { ,50},
{ptm} l = ( ,80),
{m-t|bch|cmr|pad|pmn|ppl|ptm} 013 = { ,50}, \% fl
{ptm} 013 = { ,80}, \% fl
{bch|pad|pmn} o = {50,50},
{pad|pmn} \oe = {50, },
{ppl} p = ( 0,0),
{bch|pad|pmn} q = (50,70),
{ppl} q = ( 0, ),
{m-t|cmr|pad|pmn|ppl|ptm} r = ( ,0),
{ptm} t = (50,50),
{m-t|bch|cmr|pad|pmn|ppl} y = {50,50}
{ptm} y = (80,80)
}
\SetProtrusion
{m-t} [ name = T1-sc,
{bch} [ name = bch-sc-T1,
{cmr} [ name = cmr-sc-T1,
{pad} [ name = pad-sc-T1,
{pmn} [ name = pmnj-sc-T1,
{ppl} [ name = ppl-sc-T1,
{ptm} [ name = ptm-sc-T1,
{m-t} load = T1-default ]
{bch} load = bch-T1 ]
```
CONFIGURATION FILES : Character protrusion

6574  (cmr)  load = cmr-T1 ]
6575  (pad)  load = pad-T1 ]
6576  (pnm)  load = pnm-T1 ]
6577  (ppl)  load = ppl-T1 ]
6578  (ptm)  load = ptm-T1 ]
6579  ( encoding = {T1,LY1},
6580  (bch)  family = bch,
6581  (cmr)  family = cmr,
6582  (pad)  family = {pad,padx,padj},
6583  (pnm)  family = {pnmj},
6584  (ppl)  family = {ppl,pplx,pplj},
6585  (ptm)  family = {ptm,ptmx,ptmj},
6586  shape = sc ]
6587  {
6588  a = (50,50),
6589  (cmr|pad|ppl|ptm)  \ae = {50, },
6590  (bch|pnm)  c = {50, },
6591  (bch|pad|pnm)  d = { ,50},
6592  (m-t|bch|cmr|pad|pnm|ptm)  f = { ,50},
6593  (bch|pad|pnm)  g = {50, },
6594  (m-t|cmr|pad|pnm|ppl|ptm)  j = {50, },
6595  (bch)  j = {100, },
6596  (m-t|bch|cmr|pad|pnm|ppl)  l = { ,50},
6597  (ptm)  l = { ,80},
6598  (m-t|bch|cmr|pad|pnm|ppl)  029 = { ,50}, % fl
6599  (ptm)  029 = { ,80}, % fl
6600  (bch|pad|pnm)  o = {50,50},
6601  (bch|pad|pnm)  \oe = {50, },
6602  (ppl)  p = { 0,0},
6603  (bch|pad|pnm)  q = {50,70},
6604  (ppl)  q = { 0, },
6605  (m-t|cmr|pad|pnm|ppl|ptm)  r = { ,0},
6606  t = {50,50},
6607  (m-t|bch|cmr|pad|pnm|ppl)  y = {50,50}
6608  (ptm)  y = {80,80}
6609  }
6610  }
6611  \l{!(blg|ugm)}
6612  {wm-t|cmr}
6613  \SetProtrusion
6614  (m-t)  [ name = T2A-sc,
6615  (cmr)  [ name = cmr-sc-T2A,
6616  (m-t)  load = T2A-default ]
6617  (cmr)  load = cmr-T2A ]
6618  ( encoding = T2A,
6619  (cmr)  family = cmr,
6620  shape = sc ]
6621  {
6622  \cyra = {50,50},
6623  \cyrg = { ,50},
6624  \cyrt = {50,50},
6625  \cyry = { ,50}
6626  }
6627  }
6628  {wm-t|cmr}
6629  {wm-t}
6630  \SetProtrusion
6631  (m-t)  [ name = QX-sc,
6632  load = QX-default ]
6633  ( encoding = QX,
6634  shape = sc ]
6635  {
6636  a = (50,50),
6637  f = { ,50},
6638  j = {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.
{ encoding = OT1, 
family = pmnj, 
shape = {scit,si} } 
{
a = {50, }, 
\ae = { ,-50}, 
b = {20,-50}, 
c = {50,-50}, 
d = {20, 0}, 
e = {20,-50}, 
f = {10, 0}, 
o12 = {10,-50}, % fi 
o13 = {10,-50}, % fl 
o14 = {10,-50}, % ffi 
o15 = {10,-50}, % ffl 
g = {50,-50}, 
i = {20,-50}, 
j = {20, 0}, 
k = {20, }, 
l = {20,50}, 
m = { (,-30), 
n = { (,-30), 
o = {50, }, 
\oe = {50,-50}, 
p = {20,-50}, 
q = {50, }, 
r = {20, 0}, 
s = {20,-30}, 
t = {70, }, 
u = {50,-50}, 
v = {100, }, 
w = {100, }, 
y = {50, }, 
z = { (,-50} 
} 
\SetProtrusion 
[ name = pmnj-scit-T1, 
load = pmnj-it-T1 ] 
{ encoding = {T1,LY1}, 
family = pmnj, 
shape = {scit,si} } 
{ 
a = {50, }, 
\ae = { ,-50}, 
b = {20,-50}, 
c = {50,-50}, 
d = {20, 0}, 
e = {20,-50}, 
f = {10, 0}, 
o28 = {10,-50}, % fi 
o29 = {10,-50}, % fl 
o30 = {10,-50}, % ffi 
o31 = {10,-50}, % ffl 
g = {50,-50}, 
i = {20,-50}, 
188 = {20, 0}, % ij 
j = {20, 0}, 
k = {20, }, 
l = {20,50}, 
m = { (,-30), 
n = { (,-30), 
o = {50, }, 
\oe = {50,-50}, 
p = {20,-50}, 
}
15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?
CONFIGURATION FILES: Character protrusion 178

\texttwelveudash = (200,200),
\texttwelveudash = (200,200),
\texttwelveudash = (150,150),
\texttwelveudash = (200,200),
\texttwelveudash = (500,600),
\texttwelveudash = (300,400),
\texttwelveudash = (400,500),
\texttwelveudash = (500,500),
\texttwelveudash = (300,500),
\texttwelveudash = (200,300),
\texttwelveudash = (150,200),
\texttwelveudash = (100,100),
\texttwelveudash = (50,50),
\texttwelveudash = (50,50),
\texttwelveudash = {50,50},
\texttwelveudash = {50,50},
\texttwelveudash = {50,50},
\texttwelveudash = (50,80),
\texttwelveudash = {400,400},
\texttwelveudash = {400,400},
\texttwelveudash = {400,400},
\texttwelveudash = {200,200},
\texttwelveudash = {200,200},
\texttwelveudash = {200,200},
\texttwelveudash = {200,200},
\texttwelveudash = {300,300},
\texttwelveudash = (250,300),
\texttwelveudash = (100,100),
\texttwelveudash = (100,100),
\texttwelveudash = (100,100),
\texttwelveudash = (150,150),
\texttwelveudash = (80,80),
\texttwelveudash = (200,200),
\texttwelveudash = (100,100),
\texttwelveudash = (150,150),
\texttwelveudash = (80,80),
\texttwelveudash = (100,100),
\texttwelveudash = (100,100),
\texttwelveudash = (150,150),
\texttwelveudash = (200,200),
\texttwelveudash = (400,500),
\texttwelveudash = (150,150),
\texttwelveudash = (50,100),
\texttwelveudash = (50,100),
\texttwelveudash = (50,100),
\texttwelveudash = (50,70),
\texttwelveudash = (50,50),
\texttwelveudash = (50,50),
\texttwelveudash = (50,50),
\texttwelveudash = (100,150),
\texttwelveudash = (200,200),
\texttwelveudash = (100,150),
\textsurd = \{ .100 \}

Remaining slots in the source file.

\setprotrusion
\{ cmr \} [ name = cmr-textcomp-it ]
\{ pod \} [ name = pod-textcomp-it ]
\{ pmn \} [ name = pmn-textcomp-it ]
\{ ugm \} [ name = ugm-textcomp-it ]
\{ encoding = TSI, \}
\{ cmr \} family = cmr,
\{ pod \} family = \{ pad, padx, padj \},
\{ pmn \} family = \{ pmnx, pmnj \},
\{ ugm \} family = ugm,
\{ ugm \} shape = \{ it, si \}
\{ ugm \} shape = it
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
\{ \}
15.8.6 Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from fontmath.ltx. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font ‘operators’ (also used for the \textit{\textsf{\textbf{\textnormal{\texttt{}}}}} alphabets) is OT1/cmr, which we’ve already set up above. It’s declared as:

\begin{verbatim}
\DeclareSymbolFont{operators}{OT1}{cmr}{m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
\end{verbatim}

\textit{\textsf{\textbf{\textnormal{\texttt{}}}}} is also already set up.

There are (for the moment) no settings for \textit{\textsf{\textbf{\textnormal{\texttt{}}}}} and \textit{\textsf{\textbf{\textnormal{\texttt{}}}}}.

Math font ‘letters’ (also used as \textit{\textsf{\textbf{\textnormal{\texttt{}}}}}) is declared as:

\begin{verbatim}
\DeclareSymbolFont{letters}{OML}{cmm}{m}{it}
\SetSymbolFont{letters}{bold}{OML}{cmm}{b}{it}
\end{verbatim}

\textit{\textsf{\textbf{\textnormal{\texttt{}}}}} is also already set up.

There are (for the moment) no settings for \textit{\textsf{\textbf{\textnormal{\texttt{}}}}} and \textit{\textsf{\textbf{\textnormal{\texttt{}}}}}.

Math font ‘letters’ (also used as \textit{\textsf{\textbf{\textnormal{\texttt{}}}}}) is declared as:

\begin{verbatim}
\DeclareSymbolFont{letters}{OML}{cmm}{m}{it}
\SetSymbolFont{letters}{bold}{OML}{cmm}{b}{it}
\end{verbatim}
\texttt{T = \{50,100\}, U = \{50,50\}, V = \{100,100\}, W = \{50,100\}, X = \{50,100\}, Y = \{100,100\}, f = \{100,100\}, h = \{50,100\}, i = \{50,50\}, j = \{50,50\}, k = \{50,50\}, r = \{50,50\}, v = \{50,100\}, w = \{50,100\}, x = \{50,100\},}

\texttt{\textbackslash \alpha = \{50,100\}, \beta = \{50,50\}, \gamma = \{200,150\}, \delta = \{50,50\}, \epsilon = \{50,50\}, \zeta = \{50,150\}, \theta = \{50,50\}, \iota = \{50,100\}, \kappa = \{50,100\}, \lambda = \{100,50\}, \mu = \{100,50\}, \nu = \{100,50\},}

\texttt{\chi = \{100,50\}, \psi = \{100,50\}, \omega = \{100,50\}, \varepsilon = \{100,50\}, \vartheta = \{100,50\}, \varpi = \{100,50\}, \varrho = \{100,50\}, \varsigma = \{100,50\}, \varphi = \{100,50\}, \leftharpoonup = \{100,50\}, \leftharpoondown = \{100,50\}, \rightharpoonup = \{100,50\}, \rightharpoondown = \{100,50\},}

\texttt{\lhook = \{300,200\}, \rhook = \{200,300\}, \triangleright = \{50,100\}, \triangleleft = \{50,100\}, ., \ldotp = \{500,500\}, , = \{500,500\}, < = \{100,200\}, / = \{300,400\}, > = \{200,200\}, \star = \{200,200\}, \flat = \{50,100\}, \smile = \{200,200\}, \frown = \{200,200\}, \jmath = \{100,100\}, \wp = \{100,100\}.

\texttt{\textbackslash \textbackslash Remaining slots in the source file.}
Math font 'symbols' (also used for the \texttt{\textbackslash mathcal} alphabet) is declared as:

\begin{verbatim}
\DeclareSymbolFont{symbols}{OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols}{bold}{OMS}{cmsy}{b}{n}
\end{verbatim}
We don’t bother about ‘largesymbols’, since it will only be used in display math, where protrusion doesn’t work anyway. It’s declared as:

\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}

15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).
Symbol font ‘a’.

\begin{verbatim}
7300 \{ wmsa \}
7301 \SetProtrusion
7302 \[ name = AMS-a \]
7303 \{ encoding = U,
7304 family = msa \}
7305 \{
7306 "05 = {150,250}, \% \centerdot
7307 "06 = {100,100}, \% \lozenge
7308 "07 = {50,50}, \% \blacklozenge
7309 "08 = {50,50}, \% \circlearrowright
7310 "09 = {50,50}, \% \circlearrowleft
7311 "0A = {100,100}, \% \rightleftharpoons
7312 "0B = {100,100}, \% \leftrightharpoons
7313 "0D = {-50,200}, \% \Vdash
7314 "0E = {-50,200}, \% \Vvdash
7315 "0F = {-70,150}, \% \vDash
7316 "10 = {100,150}, \% \twoheadrightarrow
7317 "11 = {100,150}, \% \twoheadleftarrow
7318 "12 = {50,100}, \% \leftleftarrows
7319 "13 = {50,80}, \% \rightrightarrows
7320 "14 = {120,120}, \% \upuparrows
7321 "15 = {120,120}, \% \downdownarrows
7322 "16 = {200,200}, \% \upharpoonright
7323 "17 = {200,200}, \% \downharpoonright
7324 "18 = {200,200}, \% \upharpoonleft
7325 "19 = {200,200}, \% \downharpoonleft
7326 "1A = {80,100}, \% \rightarrowtail
7327 "1B = {80,100}, \% \leftarrowtail
7328 "1C = {50,50}, \% \leftrightarrows
7329 "1D = {50,50}, \% \rightleftarrows
7330 "1E = {250,}, \% \Lsh
7331 "1F = {250,}, \% \Rsh
7332 "20 = {100,100}, \% \rightsquigarrow
7333 "21 = {100,100}, \% \leftrightsquigarrow
7334 "22 = {100,50}, \% \looparrowleft
7335 "23 = {50,100}, \% \looparrowright
7336 "24 = {50,80}, \% \circleq
7337 "25 = {100}, \% \succsim
7338 "26 = {100}, \% \gtrsim
7339 "27 = {100}, \% \gtrapprox
7340 "28 = {150,50}, \% \multimap
7341 "29 = {150,50}, \% \dotequaldot
7342 "2C = {100,150}, \% \trianglerighteq
7343 "2D = {100,50}, \% \precsim
7344 "2E = {100,50}, \% \lesssim
7345 "2F = {50,50}, \% \lessapprox
7346 "30 = {100,50}, \% \leqslantless
7347 "31 = {50,50}, \% \leqslantgt
7348 "32 = {100,50}, \% \curlyeqprec
7349 "33 = {50,100}, \% \curlyeqsucc
7350 "34 = {100,50}, \% \preccurlyeq
7351 "35 = {50,}, \% \lessgtr
7352 "38 = {50,50}, \% \backprime
7353 "39 = {250,250}, \% \dabar: the dash bar in \dash(left,right)arrow
7354 "3C = {50,100}, \% \succcurlyeq
7355 "3E = {100,50}, \% \preccurlyeq
7356 "40 = {50,50}, \% \sqsubset
7357 "41 = {50,}, \% \sqsupset
7358 "42 = {150,50}, \% \vartriangleleft, \lhd
7359 "43 = {150,50}, \% \vartriangleright, \rhd
7360 "44 = {100,}, \% \trianglelefteq, \unlhd
7361 "45 = {100,}, \% \trianglerighteq, \unrhd
7362 "46 = {50,100}, \% \bigstar
7363 "48 = {50,50}, \% \blacktriangledown
\end{verbatim}
"49 = { ,100}, % \blacktriangleright
"4A = [100, ], % \blacktriangleleft
"4B = { ,150}, % \dashrightarrow (the arrow)
"4C = {150, }, % \dashleftarrow
"4D = { 50, 50}, % \vartriangle
"4E = { 50, 50}, % \blacktriangle
"4F = { 50, 50}, % \triangledown
"50 = { 50, 50}, % \eqcirc
"56 = { ,150}, % \Rrightarrow
"57 = {150, }, % \Lleftarrow
"58 = {100,300}, % \checkmark
"5C = { 50, 50}, % \angle
"5D = { 50, 50}, % \measuredangle
"5E = { 50, 50}, % \sphericalangle
"5F = { , 50}, % \varpropto
"60 = {100,100}, % \smallsmile
"61 = {100,100}, % \smallfrown
"62 = { 50, }, % \Subset
"63 = { , 50}, % \Supset
"66 = {150,150}, % \curlywedge
"67 = {150,150}, % \curlyvee
"68 = { 50,150}, % \leftharpoonup times
"69 = {100, 50}, % \rightharpoonup times
"6C = { 50, 50}, % \Bumpeq
"6D = { 50, 50}, % \bump
times
"6E = {100, }, % \lll
"6F = { ,100}, % \ggg
"70 = {50,50}, % \ulcorner
"71 = {100,50}, % \urcorner
"75 = {150,200}, % \backsim
"76 = { 50,100}, % \backslash
"78 = { 50,100}, % \llcorner
"79 = {100, 50}, % \lrcorner
"7C = {100,100}, % \intercal
"7D = { 50, 50}, % \circledcirc
"7E = { 50, 50}, % \circledast
"7F = { 50, 50}, % \circleddash

Remaining slots in the source file.

Symbol font ‘b’.

\setprotrusion
\{ name = AMS-b \}
\{ encoding = U, \}
\{ family = msb \}
\{ A = { 50, 50}, % \mathbb
C = { 50, 50},
G = { , 50},
L = { , 50},
P = { , 50},
R = { , 50},
T = { , 50},
V = { 50, 50},
X = { 50, 50},
Y = { 50, 50},
00 = { 50, 50}, % \lvertneqq
01 = { 50, 50}, % \gvertneqq
02 = { 50, 50}, % \nleq
03 = { 50, 50}, % \ngeq
04 = {100, 50}, % \niess

Configuration files: Character protrusion 187
CONFIGURATION FILES: Character protrusion

7425 "05 = {50,150}, % \ngtr
7426 "06 = {100,50}, % \npreceq
7427 "07 = {50,150}, % \nsucc
7428 "08 = {100,100}, % \ngreaterslant
7429 "09 = {50,50}, % \lineq
7430 "0A = {100,100}, % \nlessapprox
7431 "0B = {50,50}, % \lneqq
7432 "0C = {50,50}, % \npreceq
7433 "0D = {50,50}, % \ngtrapprox
7434 "0E = {100,100}, % \npreceq
7435 "0F = {50,100}, % \nsuccapprox
7436 "10 = {50,50}, % \nless
7437 "11 = {50,50}, % \succapprox
7438 "12 = {50,50}, % \lineq
7439 "13 = {50,50}, % \lneq
7440 "14 = {50,50}, % \geq
7441 "15 = {50,50}, % \ngeq
7442 "16 = {100,50}, % \greaternapprox
7443 "17 = {50,50}, % \succeq
7444 "18 = {50,50}, % \precnapprox
7445 "19 = {50,50}, % \succeq
7446 "1A = {50,50}, % \lnapprox
7447 "1B = {50,50}, % \gnsim
7448 "1C = {150,200}, % \nsim
7449 "1D = {100,200}, % \nparallel
7450 "1E = {100,150}, % \nparallel
7451 "1F = {100,150}, % \nparallel
7452 "20 = {100,50}, % \|\nsubsetneq
7453 "21 = {50,100}, % \|\nsubsetneq
7454 "22 = {100,50}, % \|\nsupseteqq
7455 "23 = {50,100}, % \|\nsupseteqq
7456 "24 = {100,50}, % \|\nsupseteqq
7457 "25 = {50,100}, % \|\nsupseteqq
7458 "26 = {100,50}, % \|\nsubsetneqq
7459 "27 = {50,100}, % \|\nsubsetneqq
7460 "28 = {100,50}, % \|\supseteqq
7461 "29 = {50,100}, % \|\supseteqq
7462 "2A = {100,50}, % \|\nsupseteqq
7463 "2B = {50,100}, % \|\nsupseteqq
7464 "2C = {50,100}, % \|\nparallel
7465 "2D = {100,150}, % \|\nmid
7466 "2E = {150,150}, % \|\nshortmid
7467 "2F = {100,100}, % \|\nshortparallel
7468 "30 = {100,150}, % \|\nvDash
7469 "31 = {100,150}, % \|\nvDash
7470 "32 = {100,100}, % \|\nvDash
7471 "33 = {100,100}, % \|\nvDash
7472 "34 = {100,100}, % \|\ntrianglerighteq
7473 "35 = {100,100}, % \|\ntrianglerighteq
7474 "36 = {100,100}, % \|\ntrianglelefteq
7475 "37 = {100,100}, % \|\ntrianglelefteq
7476 "38 = {100,200}, % \|\nleftarrow
7477 "39 = {100,200}, % \|\nrightarrow
7478 "3A = {100,100}, % \|\nLeftarrow
7479 "3B = {100,100}, % \|\nRightarrow
7480 "3C = {100,100}, % \|\nLeftarrow
7481 "3D = {100,200}, % \|\nRightarrow
7482 "3E = {50,50}, % \|\ndivideontimes
7483 "3F = {50,50}, % \|\nvarnothing
7484 "60 = {200,50}, % \|\nFinv
7485 "61 = {200,50}, % \|\nGame
7486 "62 = {100,100}, % \|\eqsim
7487 "63 = {50,50}, % \|\beth
7488 "6A = {50,50}, % \|\gimel
7489 "6B = {150,50}, % \|\daleth
 CONFIGURATION FILES: Character protrusion 189

\begin{verbatim}
\"6C = {200, }, \% \lessdot
\"6D = { ,200}, \% \gtrdot
\"6E = {100,200}, \% \times
\"6F = {150,100}, \% \rtimes
\"70 = { 50,100}, \% \shortmid
\"71 = { 50, 50}, \% \shortparallel
\"72 = {200,300}, \% \smallsetminus
\"73 = {100,200}, \% \ltimes
\"74 = {50,100}, \% \thicksim
\"75 = {50,150}, \% \thickapprox
\"76 = {50,100}, \% \approxeq
\"77 = {50, 50}, \% \precapprox
\"78 = {100,100}, \% \curvearrowleft
\"79 = {50,150}, \% \curvearrowright
\"7A = {50,200}, \% \digamma
\"7B = {100, 50}, \% \varkappa
\"7F = {200, }, \% \backepsilon
\end{verbatim}

Remaining slots in the source file.

15.8.8 Euler

Euler Roman font (package euler).

\begin{verbatim}
\SetProtrusion [ name = euler ]
( encoding = U,
  family = eur )
{
  "01 = {100,100},
  "03 = {100,150},
  "06 = { ,100},
  "07 = {100,150},
  "08 = {100,100},
  "0A = {100,100},
  "0B = { , 50},
  "0C = { ,100},
  "0D = {100,100},
  "0E = { ,100},
  "0F = {100,100},
  "10 = {100,100},
  "13 = { ,100},
  "14 = { ,100},
  "15 = { , 50},
  "16 = { , 50},
  "17 = { 50,100},
  "18 = { 50,100},
  "1A = { , 50},
  "1B = { , 50},
  "1C = { 50,100},
  "1D = { 50,100},
  "1E = { 50,100},
  "1F = { 50,100},
  "20 = { , 50},
  "21 = { , 50},
  "22 = { 50,100},
  "24 = { , 50},
  "27 = { 50,100},
  "30 = {100,100},
  "31 = {100,100},
  "33 = {100,100},
  "35 = {100,100},
  "36 = {100,100},
  "37 = {100,100},
  "38 = {100,100},
  "39 = {100,100},
  "3A = {300,500},
}
\end{verbatim}
7548 "3B = \{200,400\},
7549 "3C = \{200,100\},
7550 "3D = \{200,200\},
7551 "3E = \{100,200\},
7552 A = \{\ ,100\},
7553 D = \{\ , 50\},
7554 J = \{ 50,\ },
7555 K = \{\ , 50\},
7556 L = \{\ , 50\},
7557 Q = \{\ , 50\},
7558 T = \{ 50,\ },
7559 X = \{ 50, 50\},
7560 Y = \{ 50,\ },
7561 h = \{\ , 50\},
7562 k = \{\ , 50\}
7563 }
7564
Extended by the euler-vm package.
7565 \SetProtrusion
7566 [ name = euler-vm,
7567 load = euler ]
7568 { encoding = U,
7569 family = zeur }
7570 {
7571 "28 = \{100,200\},
7572 "29 = \{100,200\},
7573 "2A = \{100,150\},
7574 "2B = \{100,150\},
7575 "2C = \{200,300\},
7576 "2D = \{200,300\},
7577 "2E = \{\ ,100\},
7578 "2F = \{100,\ },
7579 "3F = \{150,150\},
7580 "5B = \{\ ,100\},
7581 "5E = \{100,100\},
7582 "5F = \{100,100\},
7583 "80 = \{\ , 50\},
7584 "81 = \{200,250\},
7585 "82 = \{100,200\}
7586 }
7587
7588 */eur*
Euler Script font (eucal).
7589 /*eus*/
7590 \SetProtrusion
7591 [ name = euscript ]
7592 { encoding = U,
7593 family = eus }
7594 {
7595 A = \{100,100\},
7596 B = \{ 50,100\},
7597 C = \{ 50, 50\},
7598 D = \{ 50,100\},
7599 E = \{ 50,100\},
7600 F = \{ 50,\ },
7601 G = \{ 50,\ },
7602 H = \{\ ,100\},
7603 K = \{\ , 50\},
7604 L = \{\ ,150\},
7605 M = \{\ , 50\},
7606 N = \{\ , 50\},
7607 O = \{ 50, 50\},
7608 P = \{ 50, 50\},
T = { 50, 50 },
U = { 50, 50 },
V = { 50, 50 },
W = { 50, 50 },
X = { 50, 50 },
Y = { 50, 50 },
Z = { 50, 100 },
"00 = {250,250},
"18 = {200,200},
"3A = {200,150},
"40 = { 50, 100 },
"5E = {100,100},
"5F = {100,100},
"66 = { 50, 100 },
"67 = { 50, 50 },
"6E = {200,200},
00 = {250,250},
18 = {200,200},
3A = {200,150},
40 = { 50, 100 },
5E = {100,100},
5F = {100,100},
66 = { 50, 100 },
67 = { 50, 50 },
6E = {200,200},
SetProtrusion
[ name  = euscript-vm,
  load  = euscript ]
{ encoding  = U,
  family  = zeus }
{
  "01 = {600,600},
  "02 = {200,200},
  "03 = {200,200},
  "04 = {200,200},
  "05 = {150,150},
  "06 = {200,200},
  "07 = {200,200},
  "08 = {100,100},
  "09 = {100,100},
  "0A = {100,100},
  "0B = {100,100},
  "0C = {100,100},
  "0D = {100,100},
  "0E = {150,150},
  "0F = {100,100},
  "10 = {150,150},
  "11 = {100,100},
  "12 = {150,100},
  "13 = {100,150},
  "14 = {150,100},
  "15 = {100,150},
  "16 = {200,100},
  "17 = {100,200},
  "18 = {150,150},
  "19 = {150,150},
  "1A = {150,150},
  "1B = {100,150},
  "1C = {100,100},
  "1D = {100,100},
  "1E = {250,100},
  "1F = {100,250},
  "20 = {150,200},
  "21 = {150,200},
  "22 = {150,150},
  "23 = {150,150},
  "24 = {100,200},
  "25 = {150,150},
  "26 = {150,150},
  "27 = {100,100},
  "28 = {100,100},
  "29 = {100,150},
  "2A = {100,100},
CONFIGURATION FILES: Character protrusion

\begin{verbatim}
"2B = \{100,100\},
"2C = \{100,100\},
"2D = \{150,150\},
"2E = \{150,150\},
"2F = \{100,100\},
"30 = \{100,100\},
"31 = \{100,100\},
"32 = \{100,100\},
"33 = \{100,100\},
"34 = \{100,100\},
"35 = \{100,100\},
"3E = \{150,150\},
"3F = \{150,150\},
"60 = \{ 200, 200\},
"61 = \{200, 200\},
"62 = \{100,100\},
"63 = \{100,100\},
"64 = \{100,100\},
"65 = \{100,100\},
"66 = \{300, 200\},
"67 = \{300, 200\},
"68 = \{100, 100\},
"69 = \{100, 100\},
"6C = \{100,100\},
"6D = \{100,100\},
"6F = \{100,100\},
"70 = \{100,100\},
"71 = \{100,100\},
"72 = \{100,100\},
"73 = \{200,200\},
"76 = \{ 100, 100\},
"77 = \{100,100\},
"78 = \{50, 50\},
"79 = \{100,100\},
"7A = \{100,100\},
"7B = \{100,100\},
"7C = \{100,100\},
"7D = \{150,150\},
"7E = \{100,100\},
"AB = \{100,100\},
"A9 = \{100,100\},
"AB = \{200,200\},
"BA = \{200,200\},
"BB = \{ 200, 200\},
"BD = \{200,200\},
"DE = \{200,200\}
\end{verbatim}

Euler Fraktur font (eufrak).

\begin{verbatim}
\SetProtrusion
\[\text{name = mathfrak}\]
\[\text{encoding = U,}\]
\[\text{family = euf}\]}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
A = \{ 50, 50\},
B = \{ 50, 50\},
C = \{ 50, 50\},
D = \{ 80, 80\},
E = \{ 50, 50\},
G = \{ 50, 50\},
L = \{ 80, 80\},
O = \{ 80, 80\},
T = \{ 80, 80\},
X = \{ 80, 80\},
Z = \{ 80, 80\},
b = \{ 50, 50\},
c = \{ 50, 50\},
k = \{ 50, 50\},
\end{verbatim}
\end{verbatim}
15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym\[24]).

24 Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!
### 15.9 Interword spacing

Default unit is space.

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):

\parfillskip0pt
\rightskip0pt
\parfillskip\fontdimen2\font
\test test\par
\pdfadjustinterwordglue2
Some more characters in T2A.25

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’ (\fontdimen 7) for Computer Modern Roman is a third of \fontdimen 2, i.e., 333.

latex.ltx has:

\def\nonfrenchspacing{
\sfcode`\ .  3000
\sfcode`\? 3000
\sfcode`\! 3000
\sfcode`\: 2000

25 Contributed by Karl Karlsson.
15.10 Additional kerning

Default unit is 1 em.

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

15.10.1 French

The ratio of \fontdimen 2 to \fontdimen 6 varies for different fonts, so that either the kerning of the colon (which should be a space, i.e., \fontdimen 2) or that of the other punctuation characters (\TeX 's \thinspace, i.e., one sixth of \fontdimen 6) may be inaccurate, depending on which unit we choose (space or 1em). For Times, for example, a thin space would be 665. I don’t know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a
space. (Wikipedia\textsuperscript{26} claims it should be a quarter of an em, which seems too much to me; then again, it also says that this was a thin space in French typography.)

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in \textsc{pdf}\TeX .

\textbf{15.10.2 Turkish}

\texttt{\SetExtraKerning [ name = turkish, context = turkish ] { encoding = {OT1,T1,LY1} } { : = {167, }, \% = \textbackslash\text{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.

```latex
\% INHERITANCE
\% for xetex (EU1) and luatex (EU2), resp. both (TU)
\% LatinModernRoman
\DeclareCharacterInheritance
\{ encoding = {EU1,EU2,TU},
\family = Latin Modern Roman \}
\{ A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǻ,Ȁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ặ,Α}, % Greek
\AE = {Æ}, 
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ċ,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
\E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,Ε}, % Greek
\Æ = {Ǽ},
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ċ,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
\E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,Ε}, % Greek
\Æ = {Ǽ},
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ċ,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
\E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,Ε}, % Greek
\Æ = {Ǽ},
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ċ,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
\E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,Ε}, % Greek
\Æ = {Ǽ},
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ċ,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
\E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,Ε}, % Greek
\Æ = {Ǽ},
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ć,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
\E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,Ε}, % Greek
\Æ = {Ǽ},
\B = {฿, Β}, % Greek
\C = {Ç,Ć,Ĉ,Ć,Č},
\D = {Ð,Ď,Đ,Ḍ,Ḏ},
```

\[27\] This is file microtype-utf.dtx.
\[28\] Available at http://software.sil.org/charis.
\[29\] These settings have been contributed by Loren B. Davis.
### OPENTYPE CONFIGURATION FILES: Character inheritance

<table>
<thead>
<tr>
<th>Character</th>
<th>Encoding</th>
<th>Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>a</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>c</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>d</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>e</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>f</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>g</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>h</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>i</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>j</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>k</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>l</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>n</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>o</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>r</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>s</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>t</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>u</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>w</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>y</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
<tr>
<td>z</td>
<td>EU1, EU2, TU</td>
<td>Charis SIL</td>
</tr>
</tbody>
</table>

\[ \text{\LaTeX} \]
Unfortunately, I don’t have a Palatino variant containing all of the following glyphs. The settings are typeset in TeX Gyre Pagella; missing glyphs, printed in red, are taken from Charis SIL; glyphs missing even in Charis SIL appear as ’•’. To see the real settings, consult mt-PalatinoLinotype.cfg.
16.2 Character protrusion
\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\},
    \$ = \{100,-20\},
    T = \{220,-85\},
    U = \{230,-55\},
    V = \{260,-60\},
    W = \{185,-55\},
    X = \{70,-30\},
    Y = \{250,-60\},
    Z = \{90,-60\},
    a = \{150,-10\},
    b = \{170, \},
    c = \{173,-10\},
    d = \{150,-55\},
    \e = \{180, \},
    f = \{,\,250\},
    g = \{150,-10\},
    h = \{100, \},
    i = \{210, \},
    ij = \{210,-40\},
    j = \{,\,40\},
    k = \{110,-50\},
    l = \{240,-110\},
    m = \{80, \},
    n = \{115, \},
    o = \{155, \},
    q = \{170,-40\},
    r = \{155,-40\},
    s = \{130, \},
    t = \{230,-10\},
    u = \{120, \},
    v = \{140,-25\},
    w = \{98,-20\},
    x = \{65,-40\},
    y = \{130,-20\},
    z = \{110,-80\},
    \ö = \{170,-85\},
    l = \{230,110\},
<table>
<thead>
<tr>
<th>Decimal</th>
<th>Character Protrusion (in points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8366</td>
<td>2 = {130, -70}</td>
</tr>
<tr>
<td>8367</td>
<td>3 = {140, -70}</td>
</tr>
<tr>
<td>8368</td>
<td>4 = {130, 80}</td>
</tr>
<tr>
<td>8369</td>
<td>5 = {160, }</td>
</tr>
<tr>
<td>8370</td>
<td>6 = {175, -30}</td>
</tr>
<tr>
<td>8371</td>
<td>7 = {250, -150}</td>
</tr>
<tr>
<td>8372</td>
<td>8 = {130, 40}</td>
</tr>
<tr>
<td>8373</td>
<td>9 = {155, -80}</td>
</tr>
<tr>
<td>8374</td>
<td>. = {500,}</td>
</tr>
<tr>
<td>8375</td>
<td>= = {450,}</td>
</tr>
<tr>
<td>8376</td>
<td>: = {300,}</td>
</tr>
<tr>
<td>8377</td>
<td>; = {300,}</td>
</tr>
<tr>
<td>8380</td>
<td>* = {130, 30}</td>
</tr>
<tr>
<td>8381</td>
<td>% = {180, 50}</td>
</tr>
<tr>
<td>8382</td>
<td>* = {380, 20}</td>
</tr>
<tr>
<td>8383</td>
<td>+ = {180, 200}</td>
</tr>
<tr>
<td>8384</td>
<td>@ = {180, 10}</td>
</tr>
<tr>
<td>8385</td>
<td>% = {180, 50}</td>
</tr>
<tr>
<td>8386</td>
<td>/ = {100, 100}</td>
</tr>
<tr>
<td>8387</td>
<td>- = {500, 300}</td>
</tr>
<tr>
<td>8388</td>
<td>_ = {500, 300}</td>
</tr>
<tr>
<td>8389</td>
<td>= = {300, 400}</td>
</tr>
<tr>
<td>8390</td>
<td>. = {500, 300}</td>
</tr>
<tr>
<td>8391</td>
<td>. = {800, 200}</td>
</tr>
<tr>
<td>8392</td>
<td>. = {800, -20}</td>
</tr>
<tr>
<td>8393</td>
<td>' = {300, 100}</td>
</tr>
<tr>
<td>8394</td>
<td>' = {540, 100}</td>
</tr>
<tr>
<td>8395</td>
<td>' = {300, 700}</td>
</tr>
<tr>
<td>8396</td>
<td>' = {500, 300}</td>
</tr>
<tr>
<td>8397</td>
<td>' = {200, 200}</td>
</tr>
<tr>
<td>8398</td>
<td>' = {200, 200}</td>
</tr>
<tr>
<td>8399</td>
<td>/braceleft = {300, 300}</td>
</tr>
<tr>
<td>8400</td>
<td>/braceright = {200, 200}</td>
</tr>
<tr>
<td>8401</td>
<td>/C = {100, 50}</td>
</tr>
<tr>
<td>8402</td>
<td>\ = {200, -70}</td>
</tr>
<tr>
<td>8403</td>
<td>° = {220, 200}</td>
</tr>
<tr>
<td>8404</td>
<td>° = {550, 300}</td>
</tr>
<tr>
<td>8405</td>
<td>/C = {200, 70}</td>
</tr>
<tr>
<td>8406</td>
<td>\ = {200, 50}</td>
</tr>
<tr>
<td>8410</td>
<td>\ = {50, 70}</td>
</tr>
<tr>
<td>8411</td>
<td>\ = {50, 70}</td>
</tr>
<tr>
<td>8412</td>
<td>° = {140, 100}</td>
</tr>
<tr>
<td>8413</td>
<td>° = {140, 100}</td>
</tr>
<tr>
<td>8414</td>
<td>° = {400, 150}</td>
</tr>
<tr>
<td>8415</td>
<td>° = {250, 80}</td>
</tr>
<tr>
<td>8416</td>
<td>° = {250, 80}</td>
</tr>
<tr>
<td>8417</td>
<td>° = {250, 80}</td>
</tr>
<tr>
<td>8418</td>
<td>° = {300, 200}</td>
</tr>
<tr>
<td>8419</td>
<td>° = {300, 200}</td>
</tr>
<tr>
<td>8420</td>
<td>° = {200, 200}</td>
</tr>
<tr>
<td>8421</td>
<td>° = {200, 200}</td>
</tr>
<tr>
<td>8422</td>
<td>° = {150, }</td>
</tr>
<tr>
<td>8423</td>
<td>/one.oldstyle = {100, 100}</td>
</tr>
<tr>
<td>8424</td>
<td>/two.oldstyle = {100, 80}</td>
</tr>
<tr>
<td>8425</td>
<td>/three.oldstyle = {80, 50}</td>
</tr>
<tr>
<td>8426</td>
<td>/four.oldstyle = {80, 80}</td>
</tr>
<tr>
<td>8427</td>
<td>/five.oldstyle = {50, }</td>
</tr>
<tr>
<td>8428</td>
<td>/six.oldstyle = {50, }</td>
</tr>
<tr>
<td>8429</td>
<td>/seven.oldstyle = {80, 80}</td>
</tr>
<tr>
<td>8430</td>
<td>/eight.oldstyle = {50, }</td>
</tr>
</tbody>
</table>
\SetProtrusion
\setprotrusion{Charis-default}{
  \encoding{EU1,EU2,TU},
  \family{Charis SIL},
  \A={50,50},
  \AE={50,50},
  \C={50,50},
  \D={50,50},
  \E={50,50},
  \G={50,50},
  \J={100,50},
  \K={50,50},
  \L={50,50},
  \Ŀ={50,50},
  \O={50,50},
  \Œ={50,50},
  \P={50,50},
  \Q={50,50},
  \R={50,50},
  \ẞ={40,50},
  \T={50,50},
  \U={50,50},
  \V={50,50},
  \W={50,50},
  \X={50,50},
  \Y={50,50},
  \k={50,50},
  \ŀ={150,50},
  \r={50,50},
  \t={50,50},
  \u={50,50},
  \v={50,50},
  \w={50,50},
  \x={50,50},
  \y={50,50},
  \1={150,150},
  \2={50,50},
  \3={50,50},
  \4={100,50},
  \6={50,50},
  \7={50,80},
  \9={50,50},
  \.={600,500},
  \={400,300},
  \!={100,200},
  \@={50,50},
  \~={200,250},
  \%={50,300},
  *@={300,300},
}
OPENTYPE CONFIGURATION FILES: Character protrusion

+ = (200,250),
/ = (200),
\ = (150,200),
- = (400,500), % hyphen
\- = (200,300), % endash
\~ = (150,250), % emdash
\— = (100,100), % Horizontal Bar = \text{twelveudash}
\– = (200,150), % Figure Dash = \text{quartersemidash}
\– = (100,100), % HorizontalBar = \text{quartersemidash}
\– = (100,100), % FigureDash = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{twelvedash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quartersemidash}
\~ = (100,100), % Hyphen = \text{quarte
\SetProtrusion

\[ name = \text{Charis-it} \]
\( \{ \text{encoding} = \{ \text{EU1,EU2,TU} \}, \text{family} = \text{Charis SIL}, \text{shape} = \{ \text{it,sl} \} \} \)
\begin{verbatim}
8626 J = (50, ),
8627 L = (50,50),
8628 O = (50, ),
8629 Œ = (50, ),
8630 Q = (50, ),
8631 S = (50, ),
8632 S = (50, ),
8633 T = (70, ),
8634 o = (50,50),
8635 P = (50, ),
8636 q = (50, ),
8637 t = (50, ),
8638 w = (50, ),
8639 y = (50, ),
8640 I = (150,100),
8641 3 = (50, ),
8642 4 = (100, ),
8643 6 = (50, ),
8644 7 = (100, ),
8645 . = (.700),
8646 (,) = (.600),
8647 : = (.400),
8648 ; = (.400),
8649 ? = (.150),
8650 & = (.80),
8651 \% = (50,50),
8652 * = (300,200),
8653 + = (250,250),
8654 @ = (80,50),
8655 ~ = (150,150),
8656 / = (.150),
8657 /\backslash = (150,150),
8658 - = (300,400), % hyphen
8659 \( = (200,300), % endash
8660 --- = (150,200), % emdash
8661 . = (.100),
8662 ( ) = (200,200),
8663 \( = (150,200),
8664 \times = (250,250),
8665 \div = (250,250),
8666 \* = (150,200),
8667 \cdot = (300,400),
8668 \' = (400,200), = \{400,200,\}
8669 \" = (300,200), " = \{400,200,\}
8670 , = (200,500), = \{150,500,\}
8671 < = (300,400), > = (200,500),
8672 \( = (200,300), \) = \{150,400,\}
8673 \( = (200, ), \) = \{200,\}
8674 < = (200,200), > = (200,200),
8675 \braceleft = \{300,\}, \braceright = \{200,\},
8676 % Cyrillic
8677 Ж = (50,30),
8678 Л = (50, ),
8679 У = (50,30),
8680 Ф = (50, ),
8681 Ъ = (100, ),
8682 Ѳ = {50, },
8683 Ъ = {50, },
8684 Э = (50,50),
8685 Я = (50, ),
8686 Ъ = (50,50),
8687 Л = (50,50),
8688 Э = \{140,100,\},
8689 Ъ = (70,50),
8690 Л = (50,80),
\end{verbatim}
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}
% quick and dirty -- maybe we'll promote this to a regular key some time
\define@key{MT@pr@c}{command}{\csname #1\endcsname}
% glyph names have changed with version 5.0 of Charis SIL:
% before: /a.SC, /b.SC, ...
% after: /a.sc, /b.sc, ...
\ifx\MT@lua\@undefined
  \gdef\MT@get@CHARIS@SC{
    % test whether glyph "a.sc" exists
    \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
      \gdef\MT@CHARIS@SC{sc}%
    \else
      \gdef\MT@CHARIS@SC{SC}%
    \fi
  }
\else
  \gdef\MT@get@CHARIS@SC{
    \gdef\MT@CHARIS@SC{\MT@lua{
      % check font version
      \ifnum\text\print("SC")
        \gdef\MT@get@CHARIS@SC{\MT@lua{\\undefined
        \else
        \gdef\MT@get@CHARIS@SC{\\undefined
      \fi
    }}
  }
\end{verbatim}
\SetProtrusion
\[ name = Charis-sc, \]
load = Charis-default, \]
command = {MT@get@CHARIS@SC} \]
\{ encoding = {EU1,EU2,TU}, \]
family = Charis SIL, \]
shape = {sc} \}

\%
\% A = \{100,100\}, \%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\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, },
    ØE = {50, },
    P = {50, },
    Q = {50, },
    R = {50, },
    S = {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, },
    .. = {250, },
    ... = {200, },
    (,) = {500, },
    : = {300, },
    ; = {300, },
    ? = {300, },
    † = {300, },
    & = {50,50},
    \% = {100,100},
    * = {200,200},
    + = {150,200},
    @ = {50,50},
    - = {200,150},
    ( = {200, },
    ) = {200, },
    / = {100,200},
    \ = {300,500},
    \textendash = {300,300}, \textemdash = {200,200},
    \textquoteleft = {700,400}, \textquoteright = {700,400},
    \textquotedblleft = {500,300}, \textquotedblright = {500,300},
    _ = {100,100},
    \textbackslash = {100,200},
    \quotesinglbase = {500,500}, \quotationdbase = {400,400},
    \guilsinglleft = {400,400}, \guilsinglright = {300,500},
    }
\opentypeconfigurationfiles

\text{Character protrusion}

\guillemotleft = \{300,300\}, \guillemotright = \{300,300\}, \textexclamdown = \{100\}, \textquestiondown = \{200\}, \textbraceleft = \{200,100\}, \textbraceright = \{200,200\}, \textless = \{300,100\}, \textgreater = \{200,100\}, \leq = \{200,100\}, \geq = \{100,200\}, \cdot = \{850,700\}, \¶ = \{100,0\}, \× = \{150,300\}, \ª = \{300,250\}, ° = \{300,300\}, º = \{300,250\}, \⁰ = \{300,200\}, \¹ = \{300,150\}, \² = \{350,200\}, \³ = \{250,150\}, \⁴ = \{350,100\}, \⁵ = \{300,50\}, \⁶ = \{400,100\}, \⁷ = \{400,50\}, \⁸ = \{250,50\}, \⁹ = \{300,50\}, \₀ = \{300,300\}, \₁ = \{300,350\}, \₂ = \{300,150\}, \₃ = \{250,250\}, \₄ = \{400,250\}, \₅ = \{300,100\}, \₆ = \{450,200\}, \₇ = \{450,150\}, \₈ = \{125,250\}, \₉ = \{400,200\}, ± = \{150,100\}, ÷ = \{300,300\}, þ = \{50\}, † = \{250,200\}, ‡ = \{250,200\},₊ = \{300,450\},₋ = \{300,450\},⁺ = \{300,450\},⁻ = \{300,450\}, ‒ = \{300,500\}, ‑ = \{125,305\}, — = \{200,300\}, — = \{125,305\}, • = \{125,200\}

\setprotrusion

\{ name = palatino-sc, load = palatino-default \}
\encoding = \{EU1,EU2,TU\}, \family = \{PalatinoLinotype\}, \shape = \{sc\}

\{ a = \{50,50\}, æ = \{50\}, b = \{0,0\}, d = \{0,0\}, f = \{0,0\}, g = \{0,0\}, j = \{50\}, l = \{50\}, o = \{0,0\}, p = \{0,0\}, q = \{0,0\}, r = \{0,0\}, t = \{50,50\}, y = \{50,50\}, fl = \{0,50\}, ffl = \{0,50\}, ◆ = \{0,50\}, ◆ = \{0,50\} \}

\[/PalatinoLinotype\]
17 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

\begin{document}

\TestFont

Here is the beginning of a line, \dotfill and here is its end \linebreak

\end{document}
Needless to say that things may always be improved. For suggestions, mail to w.m.10@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.\(^{30}\) It has nothing to do with microtype, actually, but uses fontinst. It is based on an experiment I posted to the de.comp.text.tex newsgroup.\(^{31}\) It will show:

• the character
• the \TeX{} box
• the bounding box
• kerns

A.1 Macros

To run this file, \TeX{} needs to find the afm file (either in the TEXINPUTS path, or in the current working directory). First input fontinst.

\input fontinst.sty

bbox.sty is an addition to fontinst, which makes dimensions of the bounding boxes available (and was written by Hàn Thê´ Thành, by the way). These dimensions are specified in the afm file, but not used by \TeX{}, which is why fontinst will discard them otherwise.

\input bbox.sty

\tempdim Allocate some dimen registers.
\newdimen\tempdim
\fboxrulei Frame width of the box as \TeX{} sees it.
\newdimen\fboxrulei
\fboxrulei=0.1pt
\fboxruleii Frame width of the bounding box.
\newdimen\fboxruleii
\fboxruleii=0.1pt
\kernboxheight Height of the box indicating the kern.
\newdimen\kernboxheight
\kernboxheight=5pt
\scaletoem An auxiliary macro. Return a dimension relative to the em-width of the font. Requires e-\TeX{}.
\setcommand\scaletoem#1{\dimexpr #1 sp*\fontdimen6\font/1000\relax}

\showlogo A fontinst incantation whose sole purpose is to produce the logo. Its argument is a string (letters only).
\fontinstcc
\def\showlogo#1{%
\ifdim\fontdimen6\font = 0pt
\typeout{Warning:no-fontdimen-6-specified-\***\%\%\%\%
\***-setting-it-to-\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi=***}
\fontdimen6\font=\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi\relax
\fi
\if\fi
\installfonts
\input\metrics{}{\logofont,\metrics\printbbs{#1}\relax}

Note that the logo module will not be created when installing microtype. Instead, the source file microtype-logo.dtx is included as an attachment in the PDF file. If your PDF reader supports this, you can click here to extract it; alternatively, you may use the \pdftk tool.

Message ID: 42aa3687f0$24366$9b4ed930@newsread2.arcor-online.net
Layers.

\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}0 R
\xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
\xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
\xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
\mtl@register{canvas}
\mtl@register{characters}
\mtl@register{bounding-boxes}
\mtl@register{TeX-boxes}
\xdef\mt@order{\mt@order]}
\global\let\mtl@objects\mt@objects
\def\togglelayer#1#2{%\pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
user{/Subtype/Link
/BS << /Type/Border/W 0 >> /H/O
/A << /S/SetOCGState
/State[/Toggle \csname mtl@#1\endcsname] >>}
\pdfendlink
\}
\printbbs Preparation.
\setcommand\printbbs#1{%\setbox0\hbox{#1} %
\leavevmode \kern-\fboxrulei
\the\mtl@layer{canvas}{% \getboundarychars#1
\getboundarychars#1#2\relax %\endboundarychars #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
\vbox{color{blcolor} %
\hrule width \tempdim
height \fboxrulei} %
\}} %
\kern-\dimexpr\wd0-\scaletoem{\rpcode\font\lastchar}\relax
\printbbs #1\relax
\}
\getboundarychars Get first....
\getboundarychars#1#2\relax
\def\firstchar{`#1} %
\getlastchar#1#2\relax
\def\lastchar{`#1} %
\getlastchar #1#2\relax
\}
\getlastchar ... and last character.
\def\getlastchar#1#2{\def\firstchar{`#1} %
\def\lastchar{`#1} %
\def\lastchar #1#2\relax
\}
\printbbs #1\relax
\}
\begin{verbatim}
\ifx\relax\empty\relax\def\lastchar{``#1}\else\expandafter\getlastchar\fi #2\%
\end{verbatim}

\textbf{Loop over all characters of the string.}

\begin{verbatim}
\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}
\end{verbatim}

\textbf{Record the kern between the current and the following character, then print the character.} \texttt{\texttt{kerning}} is a fontinst command.

\begin{verbatim}
\setcommand\printbb#1#2{\setbox0\hbox{{\color{textcolor}#1}}\global\tempdim=\wd0\relax\kern-\fboxrulei}
\end{verbatim}

1. \texttt{The \TeX\ box:} Print a frame in color \texttt{textcolor}. This frame shows the glyph as \TeX\ sees it.

\begin{verbatim}
\mtl@layer{Tex-boxes}{\hbox{\lower\dimexpr\dp0+\fboxrulei\relax\hbox{\vbox{\hrule height\fboxrulei\hbox{\vrule width\fboxrulei height \dimexpr\ht0+2\fboxrulei\relax\phantom{\unhcopy0}\vrule width\fboxrulei\hrule width\fboxrulei\hrule height\fboxrulei}}}}}}
\end{verbatim}

2. \texttt{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.

\begin{verbatim}
\kern-\wd0\mtl@layer{characters}{\hbox{\box0}}
\end{verbatim}

Step back by the amount that the character's bounding box differs from the \TeX\ box on the left side.
3. The bounding box: will be printed in color bbcolor.

\mt\layer{bounding-boxes}{%
  \color{bbcolor}\
  \hbox{\lower\dimexpr-\scaletoem{\bbbottom{#1}}+\fboxruleii\relax
    \hbox{\vbox{\hrule height\fboxruleii
      \hbox to \dimexpr\scaletoem{\numexpr\bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{\vrule height \bbtop{#1}-\bbbottom{#1}\relax\%}
      \kern\dimexpr\fboxruleii+\fboxrulei\relax
    }}%}
  \setbox0=\hbox to \dimexpr\scaletoem{\numexpr\bbright{#1}-\bbleft{#1}\relax}+2\fboxruleii\relax{\vrule height \bbtop{#1}-\bbbottom{#1}\relax%}
  \kern\dimexpr\fboxruleii+\fboxrulei\relax}
%

4. The kern: We also print a small box in color kerncolor indicating the kerning between the current and the next character; filled for negative kerns, empty for positive kerns.

\kern\scaletoem{\numexpr\width{#1}-\bbright{#1}\relax}%
\mt\layer{TeX-boxes}{%
  \ifnum\thekern<0
    \color{kerncolor}\
    \kern\scaletoem{\thekern}\
    \lower\kernboxheight\hbox{\vrule width -\dimexpr\scaletoem{\thekern}\relax
      height \kernboxheight}\%}
  \else
    \color{texcolor}\
    \ifnum\thekern=0 \else
      \lower\kernboxheight\hbox{\vbox{\hrule height \fboxrulei
        \hbox{\vrule height \kernboxheight width\fboxrulei
          \kern\dimexpr\scaletoem{\thekern}-2\fboxrulei\relax
          \vrule width\fboxrulei
        }%}
      \kern\dimexpr\fboxruleii+\fboxrulei\relax
    }%}
  \fi
%}
\kern-\fboxrulei
%

\newbox\logobox
\def\printlogo{\
  \setbox0=\hbox{\vbox{\
    \kern\fboxrulei
  }%}

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{pplri8a}}}
\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{%
\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}
\togglelayer{TeX-boxes}{\TeX\ boxes}.}
\edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
\immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}
\immediate\pdfxform attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
resources {/Properties <<
\mtl@resources>> /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
\pdfannot\logodimens{%
/Subtype/Widget /FT/Btn /T(Logo)%
/F 4 % why did I say this?
/AA << /E << /SetOGState /State[/Toggle \mtl@characters] >>
/X << /SetOGState /State[/Toggle \mtl@characters] >>
/D << /SetOGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
/U << /SetOGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
>> %}
\vspace{3\baselineskip}
}

\definecolor{thered}{rgb}{0.65,0.04,0.07}
\definecolor{thegreen}{rgb}{0.06,0.44,0.08}
\colorlet{texcolor}{thegreen!50} % TeX boxes
\colorlet{kerncolor}{texcolor} % negative kerns
\colorlet{bbcolor}{thered!50} % bounding box
\colorlet{bgcolor}{black!8} % canvas
\colorlet{blcolor}{black!50} % baseline
\begin{thebibliography}{99}
\end{thebibliography}

A.2 Document

Now we can start the document.

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:

\begin{center}
\printlogo \null
\end{center}
\vfill
\expandafter\enddocument
\fi

That's it.

\fi

〈/logo〉

B The letterspacing illustration

This is \texttt{microtype-lssample.dtx}. You may treat this file in three different ways:

• compile it by itself
• \texttt{\input} it in the body of a \texttt{dtx} file
• \texttt{\input} it in the preamble: it then provides the commands
  – \texttt{\lssample}: prints the letterspacing illustration
  – \texttt{\anchorarrow}: anchors an arrow for layer (#1)
  – \texttt{\showarrow}: toggles layer (#1) or (#2), and prints (#2)

The first two cases require the style file \texttt{microtype-doc.sty}, which can be generated from \texttt{microtype.ins} with:

\makefile{microtype-doc.sty}{docsty}

\ifx\lssample\undefined
\else
\fi

Upon popular request, here's how I've created the letterspacing illustration.\footnote{Note that the \texttt{lssample} module will not be created when installing \texttt{microtype}. Instead, the source file \texttt{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 \texttt{pdftk} tool.}

B.1 Macros

Rule width and image height and depth.

\makeatletter
\newdimen\lsamount
\newdimen\lsrule
\newdimen\lsheight
\lsrule=0.2pt
\def\lsheight{8pt}
\def\lsdepth{12pt}
\makeatother

Note that the \texttt{lssample} module will not be created when installing \texttt{microtype}. Instead, the source file \texttt{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 \texttt{pdftk} tool.
Our font (Adobe Caslon).

\begin{verbatim}
\def\lsfont{\fontfamily{paca}\selectfont}

Loop over all letters in (#2), letterspacing them by (#1).
\def\dols#1#2{\lsamount=#1\relax \dolss#2\enddols}
\def\dolss#1#2\enddols{\if\empty#2\empty\divide\lsamount 2\fi \ls{#1}\if\empty#2\empty\else \dolss#2\enddols \fi}

One \texttt{tikz} picture for each letter.
\def\ls#1{\begin{tikzpicture}[remember picture,line width=\lsrule]
\mts@layer{stuff}{\node[draw=thegrey,fill=theshade,outer sep=\lsrule,anchor=base,font=\lsfont](\phantom{#1}){};}

The letter.
\node[anchor=base,font=\lsfont](#1){#1};

Two auxiliary coordinates.
\path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
\path (#1.base east) ++(-.5\lsrule,-\lsdepth) coordinate (#1R);
\mts@layer{stuff}{
Now draw the normal character width,
\draw[color=thered!75,fill=thered!30,outer sep=\lsrule](#1L) rectangle (#1R);
\ifdim\lsamount>0pt
\path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
\path (#1R) ++(\lsamount+\lsrule,\lsdepth) coordinate (#1E);
\fi}
\end{tikzpicture}
\end{verbatim}

Draw the interword space.
\def\lssp#1#2#3#4{\begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]}
\mts@layer{stuff}{\node[draw=thegreen,fill=thegreen!50,use as bounding box](#2/2,\lsdepth/2);
\coordinate(#1space) at (#2/2,\lsdepth/2);
\coordinate(#1stretch) at (#2+#3/2,+0pt);
\path (#1base east) ++(+.5\lsamount,-6pt) coordinate (#1ls);
\coordinate("{#1}shrink) at (#2-#4/2,+0pt);
\path (#1 steward) ++(\lsamount+\lsrule,*\lsdepth) coordinate (#1E);
\path (#1 steward) ++(\lsamount+\lsrule,\lsdepth) coordinate (#1E);
\path (#1 steward) ++(\lsamount+\lsrule,\lsdepth) coordinate (#1E);
\draw[color=thered!75,fill=thered!30,outer sep=\lsrule](#1L) rectangle (#1R);
\draw[color=thered,fill=thered!50,line width=0.3pt,shorten <=0.5\lsrule,->](#1E) -- (#1ls);
\draw[color=thered!50,use as bounding box](0,0) rectangle ++(+#2,+\lsdepth);
\path (#2,\lsrule) rectangle ++(+#3,-4pt+\lsrule);
\path (#2,\lsrule) rectangle ++(-#4,-4pt+\lsrule);
\path (#2,\lsrule) rectangle ++(+#3,-4pt+\lsrule);
\path (#2,\lsrule) rectangle ++(-#4,-4pt+\lsrule);
\path (#2,\lsrule) rectangle ++(+#3,-4pt+\lsrule);
\path (#2,\lsrule) rectangle ++(-#4,-4pt+\lsrule);
\draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
THE LETTERSPACING ILLUSTRATION: Macros 225

\begin{tikzpicture}
\draw[-,line width=0.3pt,shorten <=-0.5\lsrule,color=thegreen!30]
(+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt); \end{tikzpicture}

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{space}
\mts@register{stretch}
\mts@register{shrink}
\mts@register{ostretch}
\mts@register{oshrink}
\mts@register{okern}
\mts@register{ligature}
\mts@register{compatibility}
\xdef\mt@order{\mt@order]}

\anchorarrow{\text{Anchor point for the arrow in the code.}}
\newcommand\anchorarrow[1]{\tikz[\remember picture,overlay]\node(#1_c){};}
\addarrow[5][left]{\text{Add an arrow from code to image.}}
\newcommand\addarrow[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);}}}\togglelayer{\text{Toggles layer.}}
\def\toggle@layer#1#2#3{\pdfstartlink\user{/Subtype/Link /BS << /Type/Border/W 0 >> /H/O \%
/BS << /Type/Border/W 1 /S/D /D[4 1] >> \%
/C[0.7 0.7 0.7] /H/0 \%
/Contents{Click to Toggle!} /A << /S/SetOCGState /State[/Toggle \csname mts@#1\endcsname] >> \%
\rlap{#2} \%
\href{#3}{\text{\color{red}Toggle #1}}}}
\newcommand\showarrow[2][\text{Toggle layer}]{\ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi\toggle@layer{\@tempa}{\itshape #2}}
The environment for our illustration.

\def\ls@sample#1{\%
\parskip 4pt \parindent 0pt
\par
\vskip4pt
\leftskip 15pt
\mt@pseudo@marg{\color{theblue}Click on the image to show the kerns
and spacings involved. Click on emphasised words in the text below
to reveal the relation of image and code.\strut}
\mt@layer{_compatibility}{%
\mt@place{\rlap{\hskip-\marginparwidth \color{white}%
\vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}%
\mt@pseudo@marg{\color{thered}%
If you had a \acronym{PDF} viewer that understands
\acronym{PDF}, 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
\}%
\edef\ls@sample{%
\dols{0pt}{Stop}
\lssp{o}{0.45em}{0.25em}{0.15em}
\dols{0.16em}{\textit{stealing}}
\hskip-\dimexpr 0.08em+\lsrule\relax
\lssp{i}{13.82pt}{4.65pt}{2.08pt}
\dols{0.16em}{\textit{sheep}}
\dols{0pt}{!}
}%

Don’t forget to add the arrows.

\vspace{-\baselineskip}
\add@arrow{red} {tracking} {lsamount _c.east} {a_ls}
\add@arrow{red} {okern} {okernend _c.east} {p_ls}
\add@arrow{green} {ospace} {ospace _c.east} {ospace}
\add@arrow{green} {ispace} {ispace _c.center} {ispace}
\add@arrow{green!75} {istretch} {istretch_c.east} {istretch.north}
\add@arrow{green!75} {ishrink} {ishrink_c.west} {ishrink.north}
\add@arrow{green!75} {ostretch} {ostretch_c.east} {ostretch.north}
\add@arrow{green!75} {oshrink} {oshrink_c.east} {oshrink.north}
\add@arrow[right]{grey} {ligature} {nolig_c.east} {st.center}
\}
\fi
\end{miid}

This is for use with microtype.dtx
\ifx\documentclass\@twoclasseserror
\usepackage{tikz}
\else
B.2 Document
\documentclass[10pt,a4paper]{ltxdoc}
\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
Re-use the preamble from microtype.dtx.
\usepackage{microtype-doc}
\usepackage{attachfile}
\usepackage{tikz}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects]
/D << /Order [\mt@order] /BaseState/OFF >> >> }
\makeatother
\begin{document}
You are currently reading this.
\DocInput{microtype-lssample.dtx}
Now show what we are able to do.
\noindent
Since a picture is worth a thousand words, probably even more if, in our
case, it depicts a couple of letterspaced words, let's bring one to sum up
these somewhat confusing options. Suppose you had the following settings
(which I would in no way recommend; they are only for illustrative purposes):
\begin{verbatim}
\SetTracking
[ no ligatures = {"\anchorarrow{nolig}\"f},
 spacing = {60"\anchorarrow{ispace}\"0*, 
-1"\anchorarrow{istretch}\"00*, 
"\anchorarrow{ishrink}\"00*},
 outer spacing = {4"\anchorarrow{ospace}\"50, 
2"\anchorarrow{ostretch}\"50, 1"\anchorarrow{oshrink}\"50},
 outer kerning = {"\anchorarrow{okernbegin}\"*, 
"\anchorarrow{okernend}\"*} ] 
{ encoding = * }
{ 1"\anchorarrow{lsamount}\"60 }
\end{verbatim}
\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, as \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
\end{document}
C Change history

2004/09/11 Version 1.0

General: Initial version ...................... 1

2004/09/21 Version 1.1

General: configuration file names in lowercase (suggested by Harald Harders) .................. 87
remove 8-bit characters from the configuration files (suggested by Harald Harders) ............. 146
Protrusion: add factors for some more characters settings for Adobe Minion (contributed by Harald Harders) ........................................ 154
\DeclareCharacterInheritance: new command: possibility to specify character inheritance ..... 119
\DeclareDeclareSets: remove spaces around set name \MT@findfile: file: also check whether the file for the base font family has already been loaded ... 87
\MT@getbasefamily: only remove suffixes 'x' or 'j' \MT@get@listname@: don't check for empty attributes
list ............................................ 89
\ifempty: fix: use category code 12 for the percent character (reported by Tom Kink) .......... 45
\ifnumber: numbers may also be specified in hexadecimal or octal (suggested by Harald Harders) .......................................................... 94
\pdfx: fix: version check (reported by Harald Harders) .................................................. 40
\permute: don't use sets for empty encoding ... 121
\setup@expansion: issue an error instead of a warning, when pdftex version is too old for autoexpand ........................................... 136
\setup@codes: fix: allow zero and negative values \setup@set: remove spaces around set name ... 109

2004/10/03 Version 1.2

Font aliases: declare cmmr as an alias of cmr ...... 143
Font sets: new: allmath and basicmath .......... 142
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding .... 177
add settings for Computer Roman math symbols ......................................................... 182
\familyalias: define alias font name as an alternative, not as a replacement ............. 59
\get@basefamily: also remove 'w' (swash capitals) ...................................................... 88
\get@highlevel: check whether defaults have changed .................................................. 106
\get@inh@list: fix: set inheritance list \globally to \empty ..................................... 91
\get@listname@: alternatively check for alias font name ............................................. 89
\get@size: additional magic to catch some errors \hijack@set@fontsize: instead of \set@fontsize ... 107
\loop: fix: new macro, used instead of \\loop .......................... 49
\maybe@do: also check for alias font name ......................... 60
\permute@integer: more sanity checks for \set@protrusion and \expansion .................. 122
\setup@font: also search for alias font file ........ 57
\fix@catcode: check some category codes (compatibility with german) ......................... 35
\load@list: check whether list exists ............. 87

2004/10/27 Version 1.3

General: fix: specifying load option does no longer require to give a name, too ................ 116
Font aliases: declare aer, zerd and hfor as aliases of cmr ........................................ 143
\fix@catcode: check some category codes (compatibility with german) ......................... 35
\load@list: check whether list exists ............. 87

2004/11/12 Version 1.4

General: check for pdfcprot ............................. 54
don't use scratch registers in global definitions 
use \pickup@font instead of\define@newfont as the hook for \setup@font .................................... 99
use one instead of five counters ............... 50
Protrusion: tweak quote characters for cmr variants ................................................... 130
\expansion: fix: specifying extra options does no longer require to give a name, too .... 113

2004/11/17 Version 1.4a

General: new option: final ............................. 127
\cfg@catcodes: fix: reset some more catcodes when reading files (reported by Michael Hoppe) 88
**CHANGE HISTORY**

**2004/11/26**

**Version 1.4b**

General: fix: set catcodes before reading global configuration file (reported by Christoph Bier) ........................................... 129
optimisation: use less \expandafter and \csname ........................... 44
Protrusion: harmonise dashes in upshape and italic
(cm, pad, ppl) ............................................................ 153
slanted like italics ......................................................... 162
\MT@checklistfamily: fix: don’t try alias family name if encoding failed ......................... 61
\MT@getbasefamily: fix: failed for font names of the
form abczz (reported by Georg Verweyen) ............... 88
\MT@get@slot: don’t define \MT@char globally (save stack problem) .......................... 91
\MT@fdimension: don’t set \MT@count globally (save stack problem) ......................... 46
\MT@setup@PDF: new message if \pdftexoutput is changed ...................... 133
\MT@use@set: don’t use undeclared font sets .......... 109

**2004/12/15**

**Version 1.5**

General: defaults: step: 4 (suggested by Hân Thê Thành) .......................... 127
new option: selected, by default false (suggested
by Hân Thê Thành) .......................................................... 125
Documentation: add ‘Short history’ ................................... 30
Inheritance: remove \ss from T1 list, add \DJ ... 146
Protrusion: settings for Bitstream Charter ................. 154
\DeclareMicrotypeAlias: remove spaces around arguments ........................................ 110
\MT@fn@catcodes: reset catcode of ‘a’ (compatibility
with Turkish) .............................................................. 88
\MT@fix@catcodes: reset catcode of ‘î’ (compatibility
with chemsym) ............................................................. 35
\MT@get@slot: don’t test defaults if called after
begin document ........................................................... 106
\MT@scale@factor: warning for factors outside limits .......................... 65
\MT@GetFont: don’t use \upcode and \upcode for the calculation ......................... 64
\MT@set@ex@codes: allow non-selected font expansion ....................... 69
\MT@set@pro@codes: adjust protrusion factors before
setting the inheriting characters ......................... 62
\MT@setup@expansion: defaults: calculate step as
min(stretch, shrink)/5 .................................................... 135
defaults: turn off expansion for DVI output .......... 134
disable automatic expansion for DVI output .......... 135

**2005/01/24**

**Version 1.6**

General: defaults: turn off expansion for old \pdftex
versions ................................................................. 128
load a font if none is selected .......................... 56
new option: factor, by default 1000 .... 127
restructure dtx file .................................................. 142
test whether \pickup@font has changed ........... 102
test whether numeric options receive a number
use e-TgX’s \f@name and \f@defined if defined .................. 127
Protrusion: add italic uppercase Greek letters ..... 162
improve settings for numbers (pointed out by Peter
Mathusius) .............................................................. 155
\MT@fn@catcodes: reset catcode of ‘a’ (compatibility
with OML encoding) ................................................ 182
\MT@GetFont: use e-TgX’s \fontcharwd, if available ..... 64
\MT@GetFont@list: correct message if selected is
false ................................................................. 90
\MT@set@ex@codes: introduce factor option .................. 69
\MT@set@pro@codes: introduce factor option ............ 62
\MT@setup@expansion: disable automatic expansion
for old \pdftex versions ................................................ 136
\MT@use@set: retain current set if new set is un-declared ........ 109
\MT@info: new macro instead of \ifMT@verboseto 36

**2005/02/02**

**Version 1.6a**

Documentation: add table of fonts with tailored protrusion settings ...................... 21
\MT@get@slot: completely redone, hopefully more robust
(compatible with frenchpro; problem
reported by Bernard Gaulle) ........................................ 91
\MT@pdf@text@no: new macro ........................................ 39
\MT@reset@ef@codes: only reset \ef@codes for older
\pdftex versions ...................................................... 70

**2005/03/23**

**Version 1.7**

General: allow specification of size ranges (suggested
by Andreas Bühmann) .............................................. 106
\loadMicrotypeFile: new command (suggested by
Andreas Bühmann) ..................................................... 111
\newvalue{verbosename}{errors} .................................. 127
shorter command names ........................................ 51
warning when running in draft mode .................... 133
Documentation: add hint about compatibility
remove table of match order (now table 4 on page 89) ............ 26
Protrusion: fix: remove \ from OT1, add
\textbackslash to T1 encoding .................................. 156
\LoadMicrotypeFile: new command (suggested by
Andreas Bühmann) ..................................................... 111
\Microtype@Hook: new command for font package
authors ................................................................. 129
\Microtype@setup: fix: warning also when setting to
(no)compatibility .................................................. 130
\MT@begin@catcodes: also use inside configuration
commands ............................................................. 88
\MT@cfg@catcodes: reset catcode of ‘’ (compatibility
with french* packages) ........................................... 88
2005/06/23

**Version 1.8**

General: \SetProtrusion: new key: unit . 117

if font substitution has occurred, set up the substitute font, not the selected one . 99

new option: config to load a different main configuration file . 128

new option: unit, by default character . 128

Documentation: add example for factor option . 13

add example of how to get rid of a widow (suggested by Adam Kucharczyk) . 15

add hint about error messages . 27

Font aliases: declare pxr and tpx as aliases of ppl resp. ptm . 144

Font sets: add U encoding to allmath . 142

Inheritance: remove \DJ from T1 list (it's the same as \DJ) . 146

Protrusion: add LY1 characters for Times . 161

settings for AMS math macros . 185

verified settings for slanted Computer Modern Roman . 170

\addaccent: fix: disable micro-typographic setup inside \addaccent (reported by Stephan Hennig) . 101

\DeclareMicrotypeAlias: warning when overriding an alias font . 110

\DeclareMicrotypeSetDefault: new command: set default font set . 109

\MT@fp@catcodes: reset catcodes of the remaining ASCII characters . 88

\checklist: made recursive . 123

\currentlistname: new macro: current list type and name . 98

\declare@sets: warning when redefining a set . 105

\define@set@key: use comma lists instead of token lists . 105

\m@scale: new macro: use e-\TeX's \numexpr if available . 111

\set@xadd: two versions of this macro . 69

\split@name: don't define \MTencoding &c. \globally . 59

\test@ast: make it simpler . 106

\try@order: always check for size, too (suggested by Andreas Bühmann) . 89

fix: also check for \./\where\// (reported by Andreas Bühmann) . 89

\warn@code@tool@arge: new macro: type out maximum protrusion factor . 66

\warn@err: new macro: for verbose-errors . 36

\showhyphens: modify \showhyphens . 137

2005/10/28

**Version 1.9**

General: \DeclareMicrotypeSet: new key: font . 108

\SetProtrusion: value `relative' renamed to `character' for key unit . 117

allow context-specific font setup . 99

compatibility with \TeX Live hack (reported by Herbert Voß) . 39

disable microtype setup inside \hyperref's \pdfstringdef (reported by Hân Thế Thành) . 55

fix: use true as the default value . 124

option unit: rename value relative to character . 128

Documentation: add hint about \verbatim environment . 25

add remark about Type 1 fonts required for automatic font expansion . 8

Font aliases: declare qpl and qtm (qfonts, \TeX Gyre) as aliases of ppl resp. ptm . 144

Font sets: add OT4 encoding to text sets . 142

add T5 encoding to text sets . 142

Inheritance: add list for OT4 . 142

add list for T5 (requested by Hân Thế Thành) . 149
Version 1.9a

General: ‘(file name)/(line number)’ as default list name .................. 116
new option: defer setup, by default true ................................ 126
remove superfluous test whether \pchunk\font has changed ............. 102
Documentation: add explanation for error message in DVI mode ... 27
add explanation for error message with non-Type 1 fonts ............. 28
Font aliases: declare ndbch (mathdesign) as an alias of bch ........... 145
Protrusion: fix: remove ‘\ ’ from OT1 encoding .......................... 157
settings for TS encoded Charter ............................................. 153
\microtypecontext: new command: change setup context in the document .... 103
\MT@checklist@family: fix: add two missing \expandafters .................. 61
\MT@detokenize@n: fix the \TeX version ................................. 45
\MT@exp@two@n: new macros: less \expandafters ........................... 44
\MT@get@opt: new key ‘preset’ to set all characters to the specified value before loading the lists .... 67
\MT@is@active: redone; use \set@display@protect ......................... 95
\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning ........ 94
\MT@maybe@do: redone ......................................................... 60
\MT@rem@from@clist: new macro: remove an item from a comma list ........ 48
\MT@scale@factor: generalised ............................................... 65
\MT@setup@expansion: disable expansion if both step and shrink are zero .... 136
warning if user requested zero step ..................................... 135
\MT@toks: use instead of \toks0 ............................................. 42
\SetProtrusion: (et al.) new key: font .................................. 112

2005/12/05

Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR ............. 157
settings for OT4 encoding (Computer Modern Roman, Palatino, Times) .................. 153
settings for TS encoded Computer Modern Roman .................................. 153
\Disable@ligatures: new command: disable ligatures (requires pd\TeX 1.30) ........... 111
\microtypecontext: new command: change setup context in the document .......... 103
\MT@checklist@family: fix: add two missing \expandafters .................. 61
\MT@detokenize@n: fix the \TeX version ................................. 45
\MT@exp@two@n: new macros: less \expandafters ........................... 44
\MT@get@opt: new key ‘preset’ to set all characters to the specified value before loading the lists .... 67
\MT@is@active: redone; use \set@display@protect ......................... 95
\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning ........ 94
\MT@maybe@do: redone ......................................................... 60
\MT@rem@from@clist: new macro: remove an item from a comma list ............. 48
\MT@scale@factor: generalised ............................................... 65
\MT@setup@expansion: disable expansion if both step and shrink are zero .... 136
warning if user requested zero step ..................................... 135
\MT@toks: use instead of \toks0 ............................................. 42
\SetProtrusion: (et al.) new key: font .................................. 112

Version 1.9b

General: compatibility with listings: sanistise more catcodes (reported by Holger Uhr) ............ 55
compatibility with the extended@char option of the listings package ............ 55
Documentation: activate expansion in the distributed PDF .................... 1

2006/01/20

Documentation: add example of how to increase protrusion of footnote markers (suggested by Georg Verweyen) ................. 22
Protrusion: settings for URW Garamond ..................................... 154
add samples of micro-typographic features .................................. 4
\MT@features: use throughout the package to adjust to beta-ness ............. 51
\MT@f@dimen: use \pdfmatch if available .................................. 46
\MT@warn@code@too@large: fix calculation with present factor ............. 66

Version 1.9c

2006/02/02

Documentation: add example of how to increase protrusion of footnote markers (suggested by Georg Verweyen) ................. 22
Protrusion: settings for URW Garamond ..................................... 154
\MT@define@code@key@font: fix: context was ignored ..................... 115
\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in v1.9b) ................. 115

Version 1.9d

2006/05/05

Font sets: \md instead of \n series in basic sets ......................... 142
add QX encoding to text sets ............................................. 142
Inheritance: add list for QX encoding (contributed by Maciej Eder) .......... 148
Protrusion: settings for QX encoding (contributed by Maciej Eder) .......... 148
settings for Euro symbols (Adobe, ITC, marvosym) .......................... 193
tweak AMS settings ......................................................... 185
\Declare@Character@Inheritance: fix: empty context ...................... 119
\MT@detokenize@n: new macro: use \detokenize if available ............. 45
\MT@get@opt: fix: evaluate preset ....................................... 71
\MT@get@f@dimen: warning for zero f@dimen ................................ 65
\MT@get@opt: optimise: don’t reset when preset option is set ........... 67
set list name before presenting ......................................... 67
CHANGE HISTORY

<table>
<thead>
<tr>
<th>Version 1.9e</th>
<th>2006/07/28</th>
</tr>
</thead>
<tbody>
<tr>
<td>General: fix: default value for activate: true</td>
<td>125</td>
</tr>
<tr>
<td>Documentation: add hint about unknown encodings</td>
<td>26</td>
</tr>
<tr>
<td>include LPPL</td>
<td>246</td>
</tr>
<tr>
<td>Font aliases: declare zeur and zeus (eulervm) as aliases of eur resp. eus (euler)</td>
<td>145</td>
</tr>
<tr>
<td>Inheritance: adapt to marvosym's changed encoding</td>
<td>150</td>
</tr>
<tr>
<td>Protrusion: complete settings for Euler Fraktur and Script fonts</td>
<td>192</td>
</tr>
<tr>
<td>fix: forgotten comma in mt-mvs.cfg; adapt to marvosym's changed encoding</td>
<td>193</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 1.9f</th>
<th>2006/09/09</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protrusion: fix: euler-vm did not load euler settings</td>
<td>190</td>
</tr>
<tr>
<td>\SetProtrusion: (et al.) optimise: unify keys for mandatory argument</td>
<td>112</td>
</tr>
<tr>
<td>\SetProtrusion: (et al.) split keys of optional and mandatory argument</td>
<td>112</td>
</tr>
<tr>
<td>settings for Euler Roman font</td>
<td>189</td>
</tr>
<tr>
<td>\DeclareCharacterInheritance: new key 'inputenc' to set the input encoding</td>
<td>119</td>
</tr>
<tr>
<td>\SetProtrusion: model after \removelastpage</td>
<td>48</td>
</tr>
<tr>
<td>\SetSetup: empty \SetSetup after use (compatibility with the combine class)</td>
<td>52</td>
</tr>
<tr>
<td>\SetSetup: no tracing with trace package</td>
<td>101</td>
</tr>
<tr>
<td>\SetExpansion: new key: inputenc</td>
<td>113</td>
</tr>
<tr>
<td>\SetProtrusion: (et al.) set catcodes before parsing optional argument</td>
<td>112</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version 2.0</th>
<th>2007/01/14</th>
</tr>
</thead>
<tbody>
<tr>
<td>General: compatibility with \listings: set catcode of backlash to zero (reported by Steven Bath)</td>
<td>55</td>
</tr>
<tr>
<td>compatibility with soul: register \textsl and \textstyle</td>
<td>55</td>
</tr>
<tr>
<td>new option: babel, by default false (language-dependent setup suggested by Ulrich Dirr)</td>
<td>125</td>
</tr>
<tr>
<td>new option: letterspace, by default 100</td>
<td>127</td>
</tr>
<tr>
<td>new package letterspace: a stripped-down version, containing the letterspacing commands only</td>
<td>1</td>
</tr>
<tr>
<td>option 'babel': fix: switch off Turkish babel's short-hands properly (reported by Daniel Filipe)</td>
<td>141</td>
</tr>
<tr>
<td>option 'babel': switch off Turkish babel's short-hands</td>
<td>141</td>
</tr>
<tr>
<td>option 'unit', \SetProtrusion: deprecate value 'relative' completely</td>
<td>117</td>
</tr>
<tr>
<td>Documentation: add hint about how to increase font_max and font_min_size</td>
<td>28</td>
</tr>
<tr>
<td>add hint about warning when tracking and expansion is applied to a font</td>
<td>28</td>
</tr>
<tr>
<td>add remark about 'draft' option disabling microtype (noted by Michalis Matidis)</td>
<td>9</td>
</tr>
<tr>
<td>qualify hint about web documents with regard to older pdfltx versions</td>
<td>25</td>
</tr>
<tr>
<td>qualify hints about expansion error messages with regard to older pdfltx versions</td>
<td>27</td>
</tr>
<tr>
<td>Font sizes: new: footnotesize and scriptsize</td>
<td>142</td>
</tr>
<tr>
<td>new: smallcaps</td>
<td>142</td>
</tr>
<tr>
<td>\DeclareMicrotypeBabelHook: new command: interaction with babel</td>
<td>112</td>
</tr>
<tr>
<td>\liststyle: fix: font switches don't pose a problem anymore</td>
<td>78</td>
</tr>
<tr>
<td>\SetSetup: letterspacing commands may be nested</td>
<td>78</td>
</tr>
<tr>
<td>new command: letterspacing</td>
<td>78</td>
</tr>
<tr>
<td>totally redone, using the new \letterspacefont</td>
<td>78</td>
</tr>
<tr>
<td>\DeclareCharactersets: fix: empty size list when redefining set</td>
<td>105</td>
</tr>
<tr>
<td>\SetProtrusion: (et al.) set catcodes after loading input encoding (problem with listings)</td>
<td>68</td>
</tr>
<tr>
<td>\SetSetup: case 6: pdfltx 1.40</td>
<td>40</td>
</tr>
<tr>
<td>\SetSetup: no ligatures: maybe disable</td>
<td>140</td>
</tr>
<tr>
<td>\SetSetup: no ligatures after the preamble</td>
<td>140</td>
</tr>
<tr>
<td>\SetSetup: adjust to possible letterspacing</td>
<td>59</td>
</tr>
<tr>
<td>\SetExtraKerning: new command: additional kerning</td>
<td>114</td>
</tr>
<tr>
<td>\SetExtraSpacing: new command: adjustment of interword spacing</td>
<td>113</td>
</tr>
<tr>
<td>\SetTracking: new command: tracking</td>
<td>113</td>
</tr>
<tr>
<td>\textsl: new command: letterspacing</td>
<td>83</td>
</tr>
<tr>
<td>starred version: remove spaces around text</td>
<td>83</td>
</tr>
<tr>
<td>\texttt: new debug method: mark all fonts with PDF annotations</td>
<td>37</td>
</tr>
</tbody>
</table>
2007/01/21

**Version 2.1**

General: compatibility with pinyin: disable microtype in \py@macron (reported by Sven Naumann) ................. 56
fix: letterspace package forgot to load keyval ....... 41
\textls: new command: protect ligatures in letterspaced text .............. 79

2007/07/14

**Version 2.2**

General: disable microtype if wordcount is loaded (reported by Ross Hetherington) ................. 52
new option: copyfonts .................................. 126
simplify key declarations .................................. 116
use cursive trickery for e-TEx test ...................... 38
Documentation: add hint about error message with \pdfTeX 1.40.4 ................. 27
add hint about extra TOC leader dot (first discovered by Morten Høgholm) ............. 25
add overview ............................................. 5
logo transparency and amusement ....................... 1
Font aliases: declare \chr (cmath) as an alias of bch (reported by Geoff Vallis) ......... 144
declare \fp9x, \fp9j (FP\{x\}) as aliases of \ppl \{x\} ............. 144
Font sets: default set for tracking: small caps ......... 143
Inheritance: remove ‘\ ’ to ‘\27’ ............. 147
Protrusion: settings for Bitstream Letter Gothic ......... 154
Spacing: add sample .................................. 194
Tracking: add ligatures that are to be disabled ............. 150
\DisableMicrotypeVariants: new command ............. 110
\DisableLigatures: new optional argument: disable selected ligatures only ............. 111
\textls: always defined .................................. 79
\MacroCheckList\font: fix: construct font name from characteristics ............. 62
\MacroCopy\font: optionally work on copies of fonts .......... 57
\MacroGetBaseFamily: redone, working on font names and suffixes of arbitrary length ............. 143
\MacroGetChar\wd: subtract letterspacing amount from width ............. 65
\MacroGetBaseFont: fix again: remember base font in a macro ............. 79
\MacroFind\: employ Lua\TeX\ features if available ............. 46
\MacroFind: employ Lua\TeX\ features if available .......... 45
\MacroFind\: employ Lua\TeX\ features if available ............. 47
\textls: \textls version shouldn’t use \textls and \textls (found by Wiebke Petersen) ............. 47

2007/12/23

**Version 2.3**

General: disable \microtypecontext in hyperrefs \pdfstringdef ......................... 55
fix: really switch off Turkish shorthands .......... 141
new value for \il@baseoption: silent (suggested by Karl Berry) ............. 127
turned some warnings into errors .......... 127
Documentation: add kerning sample ......... 18
add letterspacing illustration ............. 17
\do@subst@correction: remember substitute font for all times (reported by Stephan Henning) ............. 101
\textls: redone: extract outer kerns from current letterspacing amount ............. 79

\microtypecontext: made robust (reported by Stephan Henning) ............. 103
\begin{catcodes}: fix: don’t disable \v@sp@df ............. 88
\define@set@key@font: font: single asterisk means normal font .......... 108
\@font\@font@list\@font\@font@list: don’t use \textls (reported by Peter Meier) ............. 48
\@font@active: support for extended Unicode (\inputenc utf8 resp. ucs) – experimental ............. 95
\@font@ligatures: fix: set evaluation didn’t work (bug introduced in \version) ............. 85
\@plain: letterspace for \eplain\minil\tics ............. 38
\@font@cur\@font@cur@os: adjusting spaces made more reliable ............. 76
2008/02/29

**Version 2.3a**

General: fix test for soul under plain \TeX{} .......................... 56
Documentation: add hint about babel having to be
looked first ........................................................................ 26
add table of available and enabled features .............. 7
mention soulutf8 .............................................................. 30
Protrusion: adjust LMX quotation marks again ... 159
Protrerror@doesn't@work: error messages if pd\TeX{} is
too old for extensions .................................................... 139
\microtypesetup: compatibility with LLMX ................... 63
\MT@fix@catcode: fix catscodes earlier, and also for the
letterspace package ......................................................... 35
\MT@get@key: fix: key=val in class options list ........... 133
\MT@set@codes: generalised ............................................ 63
\MT@set@tracking: sanity check for value .................. 113
\MT@set@tr@tracking: enable protrusion when tracking
is enabled ........................................................................ 138
\MT@set@tr@zero@: only change pre outer space if it
contains shrink ................................................................ 81

2008/06/04

**Version 2.3b**

\MT@exp@gcs: new macro: reduce save stack size .......... 44
\MT@font@copy: enable font copies also with protrusion
contexts (reported by Nathan Rosenblum) ................. 58
\MT@get@size: grouping .................................................... 107
\MT@get@oligatures@: fix: warning messages for
unknown slots .................................................................. 85
\MT@orig@pick@font: compatibility with CJKutf8: program
for debugging ................................................................ 37

2008/11/11

**Version 2.3c**

General: \LaTeX{} supported by default ...................... 40
Documentation: add hint about spacing being experi-
mental .............................................................................. 26
add hint about partial incompatibility with CJK ........ 27
Inheritance: add \textcombelow@STst@ to QX en-
coding (reported by Vasile Gaharici) ......................... 148
\MT@token@ize@gcs: fix: remove last space only (report-
ted by Ulrich Dirr) ......................................................... 45
\MT@tr@outer@r@: additional test for horizontal mode
(reporting by Sveinung Heggen) ................................ 82

2009/03/27

**Version 2.3d**

General: fix \pinyin@ compatibility check (reported by
Silas S. Brown) ............................................................... 56
move setup to the very end (for Colin Rourke) .......... 141
\MT@inannot: use pdftexcmds for debugging .......... 37
\@style: disable for Lua\TeX{} ....................................... 78
\microtypesetup: select font after setup .................... 130
\MT@check@active@set: warning for missing default
sets ................................................................................... 129
\MT@tr@ua: update for Lua\TeX{} 0.36 ......................... 41
\MT@set@tr@codes: allow zero tracking ..................... 75
\MT@set@tr@zero@: fix: allow switching off tracking
(reporting by Ulrich Dirr) .............................................. 79
\MT@setup@expansion: default step: 1 for pd\TeX{}
versions ≥ 1.40 ............................................................. 135
\MT@tr@outer@r@: don't use \textsuperscript{} (reported by
Ulrich Dirr) ................................................................. 82
\MT@tr@outer@r@: fix: don't adjust in math mode (report-
ted by Christoph Bier) ................................................... 82
\MT@tr@outer@r@: fix: don't adjust inside discretionary (reported by
Maverick Woo) .......................................................... 82
\MT@tr@set@kern: allow empty value for outer
kerning .......................................................................... 84
\text@: make math mode aware .................................... 83

2009/11/09

**Version 2.3e**

Documentation: suggest to patch \verbbatim instead
of \verbatim ................................................................. 26
Expansion: settings for T2A encoding (contributed by
Karl Karlsson) ............................................................. 196
\MT@fix@font@time@: fix: gobbling settings with
tracking failed (reported by Leo) .............................. 59
\MT@setup@: make space-unaware (requested by Mar-
cin Borkowski) .......................................................... 52
\MT@tkz@setup@: compatibility with tkz (first report-
ted by Christian Stark) ................................................ 53
\MT@tr@outer@r@: fix: set current kerning and spacing
again (found by Lars Rønnbæk) ................................. 82

\MT@set@tr@codes: also adjust tracking if protrusion is
not enabled, and even for letterspace (reported by
Stephan Hennig) ........................................................... 76
possibility to customise outer kerning (suggested by
Stephan Hennig) ........................................................... 76
\MT@Set@Tracking: sanity check for value .................. 113
\MT@setup@tracking: enable protrusion when track-
ing is enabled ................................................................ 138
\MT@tr@outer@r@: only change pre outer space if it
contains shrink .............................................................. 81
2010/01/10

Version 2.4

General: new file microtype.lua containing the lua functions (contributed by Elie Roux) .......... 43

2013/03/13

Version 2.5

General: allow contexts for LuaTeX ..... 116
disable 'Output' option for XeTeX ..... 126
fix: check whether '('file')/line') list name already exists (reported by Till A. Heilmann) ..... 116
letterspacing with LuaTeX 0.62 ..... 74
new files: microtype-pdftex.def, microtype-xetex.def, containing engine-specific definitions ..... 38
protrusion with XeTeX ..... 40
restore \space inside listings (reported by Rolf Dieterich) ..... 55
Documentation: add hint about LuaTEX compatibility ..... 27
add hint about spacing and ragged2e ..... 26
add hint about dtx source code ..... 28
include microtype-lspackage.lua and microtype-lssample.dtx ..... 218
Font aliases: declare lmsy and 1mm as aliases of cmsy resp. cmm (reported by Jonas Hogstrom) ..... 143
declare gmx etc. (garamondx) as aliases of ugm . declare Latin Modern Roman (OpenType version) as alias of 1mm when fontspec is loaded ..... 143
declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) ..... 144
Font sets: add EU1 and EU2 encodings ..... 142
Inheritance: add rudimentary list for EU1 and EU2 ..... 149
Protrusion: add default lists for EU1 and EU2 ..... 157
improvements to Computer Modern Roman italics (contributed by Hendrik Vogt) ..... 162
Tracking: add EU2 encoding to default list ..... 150
\DeclareCharacterInheritance: allow more than one encoding ..... 119
\DeclareMicrotypeAlias: ignore spaces ..... 110
\ifMT\isfamily: info if settings are not family-specific (suggested by Hân Thế Thành) ..... 62
\loadmicrotypefile: remove all spaces in font name ..... 111
\lsstyle: fix: ensure to set up math fonts (reported by RazaXsir) ..... 78

2013/05/23

Version 2.5a

General: use luatexbase instead of luatextra (contributed by Elie Roux) .......... 43
Documentation: add notes on typesetting the documentation ..... 28
include OpenType configuration files ..... 199
\MTafteraftergroup: fix: outer kerning and spacing of nested letterspacing right ..... 77
\MTget@slot@: adapt to luaotfload v2.2 (contributed by Elie Roux) .......... 93
\MT@get@slot@: fix: outer kerning and spacing of nested letterspacing right ..... 77
\MT@set@slot@: adapt to luaotfload v2.2 (contributed by Elie Roux) .......... 93
\MT@set@slot@: adapt to luaotfload v2.2 (contributed by Elie Roux) .......... 93
\MT@setupfont@hook: simplified ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmac ..... 53
\MT@ledmac@setup: support for ledmc documentation: add hint about partial incompatibility with xeCJK and luatextra (reported by Sam Mason) ..... 54
missing characters printed with Charis SIT ... 199
suggest to use etoolbox to patch \verbatim .......
Font sets: add T2 encoding (notified by Will Robertson) ...
add si and scit to smallcaps set (reported by uli) ...
new: allmath-nott and alltext-nott (requested by Karl Berry)  
Inheritance: add TU encoding  
Protrusion: add TU encoding to lists  
Tracking: add TU encoding to default list  
\DeclareMicrotypeSet: ignore spaces  
\DeclareMicrotypeSetDefault: ignore spaces  
\DeclareMicrotypeVariants: ignore spaces  
\lsstyle: fix: ensure to set up math fonts (reported by kleenstar)  
\microtypecontext: allow activate shortcut (reported by Karl Berry)  
\mt@settrcodes: use luatfont's kernfactor feature if available  
\\more: fix outer spacing problem with (not only) algorithm (reported by Henning and Ronnie Marksch)  
\\more: disable unknown slots warnings (reported by Max)  
\microtypecontext: (in)compatibility with luatexjs: disable unknown slots warnings (reported by HcN)  
compatibility with xelatex: pretend that CJK wasn't loaded  
\microtypecontext: ignore spaces  

2016/05/14 Version 2.6a
General: fixes for letterspace package with LuaTeX
\\more: fix lua function (reported by Herbert)

2017/07/ Version 2.7
General: drop luatexbase with recent l3tex ... 
warning with minimal class ... 
Documentation: mention that additional kerning does not work in math mode (discovered by 'Daniel')  
Font aliases: declare aliases for newpx ... 
declare aliases for newtx ...  
declare aliases for xcharter ...  
declare Latin Modern Roman as alias of lm with new l3tex format (reported by Ulrike Fischer)  
Protrusion: automatically choose correct names for Charis Sil small caps (reported by 'lcotoneata')  
\lsstyle: fix: prevent infinite loop with psnfss and exscale packages (reported by user11126, solution by Ulrike Fischer)  

2018/01/ Version 2.7a
General: disallow non-automatic expansion with LuaTeX  
\\more: remove 'autoexpand' for LuaTeX 1.0.6 (reported by Ulrike Fischer)  
with LuaTeX, non-automatic font expansion is no longer possible (as confirmed by Hans Hagen)  

2019/02/ Version 2.7b
General: update lua function microtype_.info after changes in luatoload (reported by Moritz Wemheuer and Ulrike Fischer)  
Documentation: update hint about non-7-bit characters (notified by Frank Mittelbach)  
Inheritance: add teextquotenoonlfigature to OT4  

INDEX

2019/10/10  Version 2.7c

General: turn warning into info when overwriting the keepligature function (reported by Andy N) ........ 96 \textls*symbol; take care of \remove@tlig ........ 86 \showhyphens; compatibility with \LaTeX{} 2019/10/01 (reported by Phylete Oleinik and Falk Hanisch) 95

2019/11/18  Version 2.7d

\MT@is@active; compatibility with \LaTeX{} 2019/10/01 from list (reported by Markus Kohm) ........ 102 \MT@is@symbol; enhance \LaTeX{} math setup, again ........ 103 \textls@activate@shortcut; fix activate shortcut ........ 103 \textls@fix@spaces; ignore spaces ........ 103 \textls@fix@fontdimen@six; try to fix zero \fontdimen@6 ........ 59 \textls@luaotf@font; use lua function ........ 49 \textls@strcmp; use \XeTeX{}'s \stricmp ........ 47 \textls@setup@expand; warning when expanding in \DVI mode with \LaTeX{} (reported by j909) ........ 134 \textls@unit@; fix: allow unit regardless whether letterspacing is set ........ 78 \textls@microtypecontext; ignore spaces ........ 103

2020/12/07  Version 2.8

General: letterspace works with \LaTeX{} only ........ 38 \textls@format; fix: enforce math setup, again ........ 79 \textls@microtypecontext; fix activate shortcut ........ 103 \textls@fix@Spaces; ignore spaces ........ 103 \textls@fix@fontdimen@six; try to fix zero \fontdimen@6 ........ 59 \textls@luaotf@font; use lua function ........ 49 \textls@strcmp; use \XeTeX{}'s \stricmp ........ 47 \textls@setup@expand; warning when expanding in \DVI mode with \LaTeX{} (reported by j909) ........ 134 \textls@unit@; fix: allow unit regardless whether letterspacing is set ........ 78 \textls@microtypecontext; ignore spaces ........ 103

D  Index

Numbers in upright shape refer to the page where the corresponding entry is described (bold face) resp. occurs. Numbers in italics refer to the code line where the corresponding entry is defined (underlined) resp. used.

Options

DVoutput ............................................... 9 \letterspace ........................................... 8 activate ............................................... 6 \protrusion ............................................ 6 auto .................................................. 7 \selected .............................................. 8 babel .................................................. 9 \shrink ............................................... 8 config .................................................. 9 \spacing ............................................... 6 draft .................................................. 9 \step .................................................. 8 expansion ........................................... 6 \stretch .............................................. 8 factor .................................................. 7 \tracking .............................................. 7 final .................................................. 9 \unit ................................................... 7 kerning ............................................... 6 \verbose .............................................. 9

Commands

\DeclareCharacterInheritance .................................. 20 \SetExtraSpacing ........................................... 19 \DeclareMicrotypeAlias ..................................... 21 \SetProtrusion .......................................... 13 \DeclareMicrotypeBabelHook ................................ 23 \SetTracking ............................................. 15 \DeclareMicrotypeSet ...................................... 10 \UseMicrotypeSet ........................................ 12 \DeclareMicrotypeSetDefault ................................ 10 \\slig .................................................. 24 \\style .................................................. 23 \\textls .............................................. 23 \\textls* .............................................. 23 \\textls@setup .......................................... 22 \\textls@microtypecontext ................................ 22

\DeclareMicrotypeVariants .................................. 20 \\microtypecontext ...................................... 9 \\loadmicrotypefile ..................................... 22 \\textls* .............................................. 23 \\textls .............................................. 23 \\setexpansion .......................................... 14 \\textls@microtypecontext ................................ 22

\SetExtraKerning ............................................ 18
The \LaTeX Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 \LaTeX3 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 \TeX-related (such as a \LaTeX package), but it is written in such a way that you can use it even if your work is unrelated to \TeX.

The section 'WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE', below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

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.

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the \LaTeX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document ‘\modguide.tex’ and ‘\cfgguide.tex’ in the base \LaTeX distribution for suggestions.

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 us-
ing installation facilities provided by the Work, trans-
formations 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 nom-
ination 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 compon-
ents but these are not considered part of the Base
Interpreter provided that each external component
clearly identifies itself whenever it is used interact-
ively. Unless explicitly specified when applying the
license to the Work, the only applicable Base Inter-
preter is a ‘\LaTeX-Format’ or in the case of files belong-
ing to the ‘\LaTeX-Format’ 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 creat-
ing a Derived Work. You may also distribute the De-
vised Work without restriction, including Compiled
Works generated from the Derived Work. Derived
Works distributed in this manner by the Current Main-
tainer 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 fol-
lowing 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 In-
terpreter, then, wherever this component of the
Work identifies itself to the user when used inter-
actively with that Base Interpreter, the replace-
ment component of this Derived Work clearly and
unambiguously identifies itself as a modified ver-
sion 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 refer-
ence 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 au-
thors of the original version of the Work, provide
any support, including (but not limited to) the re-
porting and handling of errors, to recipients of the
Derived Work unless those persons have stated
explicitly that they do provide such support for
the Derived Work.

(d) You distribute at least one of the following with
the Derived Work:

i. A complete, unmodified copy of the Work; if
your distribution of a modified component is
made by offering access to copy the modified
component from a designated place, then of-
ferring equivalent access to copy the Work from
the same or some similar place meets this con-
dition, even though third parties are not com-
pelled to copy the Work along with the modi-
fied component;

ii. Information that is sufficient to obtain a com-
plete, 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 dis-
tributed 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 ap-
plying 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 dif-
ferent license provided that license itself honors
the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.

(b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.

12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work 'as is', without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status 'author-maintained' if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is 'author-maintained'.

The Work has the status 'maintained' if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status 'maintained' to 'unmaintained' if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.

2. If this search is successful, then enquire whether the Work is still maintained.

(a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.

(b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a \LaTeX work, this could be done, for example, by posting to comp.text.tex.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.

(b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.

4. If you make an 'intention announcement' as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.
Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as 'you' in this section.

Choosing This License or Another License

If for any part of your work you want or need to use distribution conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document `modguide.tex' in the base \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:

```
\% This work consists of all files listed in manifest.txt.
```

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 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
\% and 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.

% This work consists of all files listed in manifest.txt.