The microtype package

Subliminal refinements towards typographical perfection

— IMPLEMENTATION —

R Schlicht
w.m.l@gmx.net

v3.2
2024/12/12

https://github.com/schlcht/microtype

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

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

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

Contents

List of Tables

List of Figures

User manual (external document)

1 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`).

`show`: The code for the `microtype-show` package (`microtype-show.sty`).

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

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

`xe-`: Definitions specific to XƎTEX (`microtype-xetex.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`).
  - `ebg`: Settings for EB Garamond (`mt-EBGaramond.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`).
  - `mvs`: Settings for `marvosym` Euro symbol (`mt-mvs.cfg`).

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

And now for something completely different.

```
⟨*package|letterspace⟩
```

1.1 Preliminaries

\MT@MT

This is us.

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

\MT@fix@catcode

\MT@restore@catcodes

We have to make sure that the category codes of some characters are correct (the german package, for instance, makes " active). Probably overly cautious. Ceterum censeo: it should be forbidden for packages to change catcodes within the preamble. Polite as we are, we'll restore them afterwards.

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

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

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

```
⟨*package⟩
\newcommand*\DeclareMicrotypeSet[3][]{}
\newcommand*\UseMicrotypeSet[2][]{}
\newcommand*\DeclareMicrotypeSetDefault[2][]{}
\newcommand*\SetProtrusion[3][]{}
\newcommand*\SetExpansion[3][]{}
\newcommand*\SetTracking[3][]{}
\newcommand*\SetExtraKerning[3][]{}
\newcommand*\SetExtraSpacing[3][]{}
\newcommand*\DisableLigatures[2][]{}
\newcommand*\DeclareCharacterInheritance[3][]{}
\newcommand*\DeclareMicrotypeVariants[1]{}
\newcommand*\DeclareMicrotypeAlias[2]{}
\newcommand*\LoadMicrotypeFile[1]{}
\newcommand*\DeclareMicrotypeFilePrefix[1]{}
\newcommand*\DeclareMicrotypeBabelHook[2]{}
\newcommand*\microtypesetup[1]{}
\newcommand*\microtypecontext[1]{}
\newcommand*\textmicrotypecontext[2]{#2}
\newcommand\leftprotrusion[1]{#1}
\newcommand\rightprotrusion[1]{#1}
\providecommand*\noprotrusion{}
\newcommand*\noprotrusionifhmode{}
```

```
\@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
⟨/package⟩
\newcommand*\lsstyle{}
\newcommand\textls[2][]{}
\def\textls#1#{}
\newcommand*\lslig[1]{#1}
⟨*package⟩
}
```

These commands also have a starred version.

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

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

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

Don’t load `letterspace`.

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

`\MT@old@cmd`

The old command names had one more hunch (`\..MicroType..`). Before finally letting them sink into oblivion, raise an error.

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

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

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

Communicate.

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

1.1.1 Debugging

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

Cases for `\tracingmicrotype`:

0: almost none

1: + sets & lists

2: + heirs

3: + slots

4: + factors

```
⟨*debug⟩
\MT@warning@nl{This is the debug version}
\newcount\tracingmicrotype
\tracingmicrotype=2
\def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
\def\MT@info@nl#1{\PackageInfo\MT@MT{#1\@gobble}\MT@addto@annot{#1}}
\let\MT@vinfo\MT@info@nl
\def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
\def\MT@warning@nl#1{\PackageWarning\MT@MT{#1\@gobble}\MT@addto@annot{Warning: #1}}
\def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
\def\MT@dinfo@nl#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@nl{#2}\fi}
```

`\tracingmicrotypeinpdf`

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

Cases for `\tracingmicrotypeinpdf`:

1: show new fonts

2: + show known fonts

```
\newcount\tracingmicrotypeinpdf
```

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

```
\tracingmicrotypeinpdf=2
```

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

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

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

`\iftracingmicrotypeinpdfall`

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

```
\newif\iftracingmicrotypeinpdfall
```

`\MT@show@pdfannot`

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

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

1.1.2 Visual debugging

The microtype-show package offers some tools for preparing protrusion settings. We make use of the microtype infrastructure, redefining some of its internal commands (done later, in sections 1.2.1 and 1.2.8). First, some preparation:

```
⟨*show⟩
\RequirePackage{iftex}
\ifetex\else
  \PackageError{microtype-show}
               {This package only works with e-TeX}{Use e-TeX}
\fi
\ifxetex
  \PackageError{microtype-show}
               {This package only works with pdfTeX or luaTeX}{Don't use XeTeX}
\fi
\PackageWarning{microtype-show}{DO NOT USE THIS PACKAGE FOR REAL DOCUMENTS\@gobble}
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{microtype}}
\ProcessOptions\relax
\PassOptionsToPackage{verbose}{microtype}
\RequirePackage{microtype,graphicx,xcolor}
```

\ifShowGlyphIndex
\ifShowMissingGlyphs
\GlyphScaleFactor
\Showbaselinecolor
\Showposcolor
\Shownegcolor

The following commands are configurable:

```
\newif\ifShowGlyphIndex
\newif\ifShowMissingGlyphs
\newcommand*\GlyphScaleFactor{2}
\newcommand*\Showbaselinecolor{\color{black!40}}
\newcommand*\Showposcolor{\color{green!50}}
\newcommand*\Shownegcolor{\color{red!50}}
```

\MTS@printtext
\MTS@show@index
\MTS@crulefill

Make sure to have a readable font.

```
\ifluatex
  \def\MTS@printtext#1{{\usefont{TU}{lmr}{m}{n}#1}}
\else
  \def\MTS@printtext#1{{\usefont{T1}{cmr}{m}{n}#1}}
\fi
\def\MTS@show@index#1{\ifShowGlyphIndex{\tiny$_{#1}%
 %  \ifluatex^{\mathrm{%
 %    \MT@lua{tex.print(luaotfload.aux.name_of_slot(tonumber([[#1]])))}}}\fi
  $}\fi\space}
\def\MTS@crulefill{\leaders\hrule height \dimexpr1ex/2+.4pt depth -\dimexpr1ex/2\hfill}
```

\MTS@Prot
\MTS@Char

Add the show commands to microtype's setup.

```
\g@addto@macro\MT@setupfont{\MTS@Prot\MTS@Char}
\let\MTS@Prot\relax
\let\MTS@Char\relax
```

\MTS@setup
\MTS@glyphlist

Common setup. \MTS@glyphlist stores all glyphs we've seen.

```
\def\MTS@setup{%
  \fboxsep=0pt
  \fboxrule=.1pt
  \raggedright
  \let\MTS@glyphlist\@gobble
  \def\MT@feat{pr}%
}
```

\ShowProtrusion

Activate the sleeper command, then trigger the setup.

```
\newcommand*\ShowProtrusion{%
  \begingroup
    \MTS@setup
    \let\MTS@Prot\MTS@Prot@do
    \def\MT@cat{c}%
    \selectfont
}
```

\MTS@Prot@do

But in all other cases of a font being picked up, there should be no special treatment. After we're done, select the previous font again.

```
\def\MTS@Prot@do{%
    \MT@ltx@pickupfont
    \let\MT@pr@split@val\MTS@pr@split@val
    \let\MT@load@list\MTS@load@list
    \let\MT@set@pr@prefixes@\MTS@set@pr@prefixes@
    \MTS@show@pr
  \endgroup
  \aftergroup\selectfont
}
```

\ShowCharacterInheritance

```
\newcommand*\ShowCharacterInheritance{%
  \begingroup
    \MTS@setup
    \let\MTS@Char\MTS@Char@do
    \def\MT@cat{inh}%
    \selectfont
}
```

\MTS@Char@do

```
\def\MTS@Char@do{%
    \MT@ltx@pickupfont
    \let\MT@set@pr@prefixes@\MTS@set@pr@prefixes@
    \MTS@show@inheritance
  \endgroup
  \aftergroup\selectfont
}
```

\ShowProtrusionLineGlyph

By glyph.

```
\newcommand*\ShowProtrusionLineGlyph[1]{%
  {\MTS@setup
   \MTS@showprotrusionline{`#1}}%
}
```

\ShowProtrusionLineIndex

By glyph number.

```
\newcommand*\ShowProtrusionLineIndex[1]{%
  {\MTS@setup
   \MTS@showprotrusionline{#1}}%
}
```

\MTS@showprotrusionline
\MTS@lpcode
\MTS@rpcode

```
\def\MTS@showprotrusionline#1{%
  \edef\MTS@lpcode{\number\lpcode\font#1}%
  \edef\MTS@rpcode{\number\rpcode\font#1}%
  \char#1%
    lorem ipsum dolor sit amet, \MTS@crulefill\ %
    \MTS@printtext{\ifnum\MTS@lpcode=\z@\Showbaselinecolor\fi[\MTS@lpcode]}
    \fbox{\char#1}\MTS@show@index{\number#1}
    \MTS@printtext{\ifnum\MTS@rpcode=\z@\Showbaselinecolor\fi[\MTS@rpcode]}
    \MTS@crulefill\ you know the rest%
  \char#1\par
  \ShowDummyLine
}
```

\ShowDummyLine

The first and last glyphs in this line should have a straight (non-protruded) shape. We also reset to default shape and series, because that's what, say, italic shapes should be matched with.

```
\newcommand*\ShowDummyLine{%
 {\fontencoding{\encodingdefault}\fontseries{\seriesdefault}\fontshape{\shapedefault}%
  \selectfont\noindent
  here is the beginning of a line, \dotfill and here is its end}\par
}
```

\ShowProtrusionAll

```
\newcommand*\ShowProtrusionAll{%
  {\MTS@setup
   \MTS@lede{}%
   \MT@do@font{\iffontchar\font\@tempcnta\MTS@showprotrusionline{\@tempcnta}\fi}}%
}
```

\ShowProtrusionDefined

```
\newcommand*\ShowProtrusionDefined{%
  {\MTS@setup
   \MTS@lede{defined}%
   \let\MTS@first\@gobble
   \let\MTS@second\@firstofone
   \MT@do@font{%
     \MTS@firstorsecond
     \MTS@temp{%
       \iffontchar\font\@tempcnta\MTS@showprotrusionline{\@tempcnta}\else
         \MT@warning@nl{Glyph \the\@tempcnta\space is missing in font
                        \MessageBreak\font@name}%
       \fi}}}%
}
```

\ShowProtrusionMissing

```
\newcommand*\ShowProtrusionMissing{%
  {\MTS@setup
   \MTS@lede{missing}%
   \let\MTS@first\@firstofone
   \let\MTS@second\@gobble
   \MT@do@font{%
     \MTS@firstorsecond
     \iffontchar\font\@tempcnta\MTS@temp{\MTS@showprotrusionline{\@tempcnta}}\fi}}%
}
```

\MTS@lede

```
\def\MTS@lede#1{%
  \selectfont
  \edef\MTS@font{\expandafter\string\font@name}%
  \MTS@printtext{All glyphs \MT@ifempty{#1}{in}{#1 in protrusion list for}
                 font \texttt{\MTS@font}:}\par
  \ShowDummyLine
}
```

\MTS@firstorsecond

```
\def\MTS@firstorsecond{%
  \let\MTS@temp\MTS@first
  \ifnum\lpcode\font\@tempcnta=\z@ \else
    \let\MTS@temp\MTS@second
  \fi
  \ifnum\rpcode\font\@tempcnta=\z@ \else
    \let\MTS@temp\MTS@second
  \fi
}
```

\MTS@charwd
\MTS@lp@
\MTS@rp@
\MTS@show@char@pr

Display the glyph with protrusion.

```
\newdimen\MTS@charwd
```

```
\newdimen\MTS@lp@
\newdimen\MTS@rp@
\def\MTS@show@char@pr#1{%
  \xdef\MTS@glyphlist{\MTS@glyphlist,#1}%
  \scalebox{\GlyphScaleFactor}{\strut\escapechar`\\
    \MTS@charwd=\fontcharwd\MT@font#1\relax
```

The baseline rule.

```
    {\Showbaselinecolor\vrule width \dimexpr\MTS@charwd+.3em\relax height 1sp depth 0pt}%
    \hskip-\dimexpr\MTS@charwd+.15em\relax
```

Left protrusion.

```
    {\ifdim\MTS@lp@<\z@\Shownegcolor\else\Showposcolor\fi
     \vrule width \ifdim\MTS@lp@<\z@ -\fi\MTS@lp@ height 1em depth .2em}%
    \hskip\dimexpr\MTS@charwd\ifdim\MTS@lp@>\z@-\MTS@lp@\fi
                             \ifdim\MTS@rp@>\z@-\MTS@rp@\fi\relax
```

Right protrusion.

```
    {\ifdim\MTS@rp@<\z@\Shownegcolor\else\Showposcolor\fi
     \vrule width \ifdim\MTS@rp@<\z@ -\fi\MTS@rp@ height 1em depth .2em}%
    \hskip-\dimexpr\MTS@charwd+\fboxrule\ifdim\MTS@rp@<\z@-\MTS@rp@\fi\relax
```

Finally the glyph, so that it’s on top.

```
    \fbox{\char#1}}\,%
    \MTS@show@index{#1}%
}
```

\MTS@show@char

\MTS@show@char@x

Just show the glyph; the second command also remembers it.

```
\def\MTS@show@char#1{\scalebox{\GlyphScaleFactor}{%
  \strut\fbox{\char#1}}\MTS@show@index{#1}}
\def\MTS@show@char@x#1{\xdef\MTS@glyphlist{\MTS@glyphlist,#1}\MTS@show@char{#1}}
```

\MTS@show@missing

```
\def\MTS@show@missing{%
  \MT@ifdefined@c@T\MT@pr@inh@name{%
    \MTS@lp@=\z@ \MTS@rp@=\z@
    \par \MTS@printtext{Glyphs not included in configuration (with defined heirs):}%
    \MT@do@font{%
      \edef\MT@temp{\the\@tempcnta}%
      \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @\MT@temp @}{%
        \MT@exp@one@n\MT@in@clist\MT@temp\MTS@glyphlist
        \ifMT@inlist@\else \newline
        \llap{\MTS@show@char@pr{\MT@temp} \MTS@printtext{=} }%
         \MT@exp@cs\MT@map@tlist@c
           {MT@inh@\MT@pr@inh@name @\the\@tempcnta @}%
           \MTS@show@char@x
        \fi
      }%
    }%
  }%
  \MTS@show@missing@
}
```

\MTS@show@missing@

```
\def\MTS@show@missing@{%
  \par \MTS@printtext{Other glyphs not in configuration:}\newline
  \MT@do@font{%
    \edef\MT@temp{\the\@tempcnta}%
    \MT@exp@one@n\MT@in@clist\MT@temp\MTS@glyphlist
    \ifMT@inlist@\else
      \MTS@show@char\MT@temp
    \fi
  }%
}
```

\MTS@show@inheritance

```
\def\MTS@show@inheritance{%
  \MT@get@inh@list
  \MTS@printtext{Character inheritance for font `\texttt{\MT@@font}':}\\
  \MT@ifdefined@c@TF\MT@listname{%
    \MTS@printtext{First matching list is for `\texttt{\@tempa}':\\
              \texttt{\MT@listname}:}\par\leavevmode
    \MT@do@font{%
      \MT@ifdefined@n@T{MT@inh@\MT@listname @\the\@tempcnta @}{%
        \newline
        \xdef\MTS@glyphlist{\MTS@glyphlist,\the\@tempcnta}%
        \llap{\MTS@show@char{\the\@tempcnta}\MTS@printtext{= }}%
        \MT@exp@cs\MT@map@tlist@c
          {MT@inh@\MT@listname @\the\@tempcnta @}%
          \MTS@show@char@x
      }%
    }%
    \MT@ifdefined@n@T{MT@inh@\MT@listname @prefixes}{%
      \par \MTS@printtext{(with prefixes:)}%
      \@tempcntb=\z@
      \let\MTS@show@char@pr\MTS@show@char@x
      \MT@set@pr@prefixheirs}%
    \ifShowMissingGlyphs\MTS@show@missing@\fi
  }{%
    \MTS@printtext{NOT DEFINED}%
  }%
  \par
}
⟨/show⟩
```

1.1.3 Requirements

Back to the user packages.

\MT@plain

The letterspace package works with:

0: miniltx

1: eplain

2: LaTeX

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

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

\MT@requires@latex

Better use groups than plain ifs.

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

For definitions that depend on e-TeX features.

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

We check whether we are running pdfTeX, XeTeX, or LuaTeX, and load the appropriate definition file (later in section 1.4.2).

\MT@clear@options

If we are using neither of these engines, or a too old version, we disable everything and exit.

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

A hack circumventing the TeX Live 2004 hack which undefines the pdfTeX primitives in the format in order to hide the fact that pdfTeX is being run from the user. This has been *fixed* in TeX Live 2005.

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

\MT@engine
\ifMT@engine@unfit
\MT@engine@minversion

Old packages might have let `\pdftexversion` to `\relax`.

```
\let\MT@engine\relax
\newif\ifMT@engine@unfit
\MT@engine@unfittrue
\ifx\pdftexversion\@undefined \else
  \ifx\pdftexversion\relax \else
    \def\MT@engine{pdf}
⟨package⟩    \def\MT@engine@minversion{0.14f}
⟨letterspace⟩    \let\MT@pdf@or@lua\@firstoftwo
    \ifnum\pdftexversion
⟨package⟩        > 13
⟨letterspace⟩        > 139
      \MT@engine@unfitfalse
⟨package⟩      \ifnum \pdftexversion=14
⟨package⟩        \ifnum \expandafter`\pdftexrevision < `f
⟨package⟩          \MT@engine@unfittrue
⟨package⟩        \fi
⟨package⟩      \fi
    \fi
  \fi
\fi
```

```
\ifx\directlua\@undefined \else
  \ifx\directlua\relax \else
    \def\MT@engine{lua}
    \MT@engine@unfitfalse
```

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

```
⟨*letterspace⟩
    \let\MT@pdf@or@lua\@secondoftwo
    \ifnum\luatexversion < 62 \MT@engine@unfittrue
    \else
      \let\MT@lua\directlua
      \ifnum\luatexversion > 84
        \let\pdfoutput\outputmode
        \let\pdfprotrudechars\protrudechars
        \let\pdfadjustspacing\adjustspacing
      \fi
    \fi
⟨/letterspace⟩
  \fi
\fi
⟨*package⟩
\ifx\MT@engine\relax
  \ifx\XeTeXversion\@undefined \else
    \ifx\XeTeXversion\relax \else
      \def\MT@engine{xe}
      \def\MT@engine@minversion{0.9997}
      \ifdim 0\XeTeXrevision pt > 0.9996pt
        \MT@engine@unfitfalse
      \fi
    \fi
  \fi
\fi
⟨/package⟩
⟨/package|letterspace⟩
```

`\MT@pdftex@no`

pdfTEX's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdfTEX we're using, if any. `\MT@pdftex@no` will be used throughout the package to respectively do the right thing. Currently, we have to distinguish the following cases for pdfTEX:

0: not running pdfTEX

1: pdfTEX (< 0.14f) (already checked above)

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

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

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

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

6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`[1]; `\pdftracingfonts`; always e-TEX (≥ 1.40)

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

8: + `\letterspacefont` uses explicit `\fontdimen` 6 if specified (≥ 1.40.23)

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

```
⟨*pdf-⟩
⟨debug⟩\MT@dinfo@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
\def\MT@pdftex@no{8}
\ifnum\pdftexversion = 140
  \ifnum\pdftexrevision < 23
    \def\MT@pdftex@no{7}
    \ifnum\pdftexrevision < 4
      \def\MT@pdftex@no{6}
    \fi
  \fi
\else
  \ifnum\pdftexversion < 140
    \def\MT@pdftex@no{5}
    \ifnum\pdftexversion < 130
      \def\MT@pdftex@no{4}
      \ifnum\pdftexversion < 120
        \def\MT@pdftex@no{3}
        \ifnum\pdftexversion = 14
          \ifnum \expandafter`\pdftexrevision < `h
            \def\MT@pdftex@no{2}
          \fi
        \fi
      \fi
    \fi
  \fi
\fi
⟨debug⟩\MT@dinfo@nl{0}{pdftex no.: \MT@pdftex@no}
⟨/pdf-⟩
```

`\MT@xetex@no` XƎTeX supports character protrusion since version 0.9997. This test is not necessary here, we just keep it for the (unlikely) case that features get added to XƎTeX in the future.

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

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

0: N/A

1: LuaTeX (< 0.36)

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

3: + `\letterspacefont`; non-automatic expansion doesn't work anymore, and automatic expansion in DVI mode is realised by modifying the tracking, not the glyphs[2] (≥ 0.62)

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

5: + default `\efcode` = 1000; `\protrusionboundary` [doesn't seem to work] (≥ 0.90)

6: + `\glet`(≥ 1.10)

Also, sometime between 1.0.4 and 1.0.7, the function `font.setexpansion` has been introduced (but we're not using it for now).

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

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

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

```
\let\MT@lua\directlua
\def\MT@luatex@no{6}
\ifnum\luatexversion<110
  \def\MT@luatex@no{5}
  \ifnum\luatexversion<90
    \def\MT@luatex@no{4}
    \ifnum\luatexversion<85
      \def\MT@luatex@no{3}
      \ifnum\luatexversion<62
        \def\MT@luatex@no{2}
        \ifnum\luatexversion<36
          \def\MT@lua{\directlua0}
          \def\MT@luatex@no{1}
        \fi
      \fi
    \fi
  \fi
\fi
```

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

Abort if no capable engine found.

```
⟨*package|letterspace⟩
\ifMT@engine@unfit
  \MT@warning@nl{You
    \ifx\MT@engine\relax
      don't seem to be using pdftex%
⟨package⟩      , luatex or xetex%
⟨letterspace⟩      \space or luatex%
     .\MessageBreak `\MT@MT' only works with these engines.%
    \else
      are using a \MT@engine tex version older than
⟨package⟩       \MT@engine@minversion
⟨letterspace⟩      \MT@pdf@or@lua{1.40}{0.62}%
     .\MessageBreak `\MT@MT' does not work with this version.%
      \MessageBreak Please install a newer version of \MT@engine tex.%
    \fi
    \MessageBreak I will quit now}
  \MT@clear@options
\endinput\fi
```

Still there? Then we can begin: We need the `keyval` package, including the 'new' `\KV@@sp@def` implementation. For the `patch` option, we use `etoolbox`, which requires e-TeX.

```
\RequirePackage{keyval}[1997/11/10]
⟨*package⟩
^^X\RequirePackage{etoolbox}
\providecommand\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
```

\MT@toks We need a token register,

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

\MT@tempbox our own box,

```
\newbox\MT@tempbox
```

\ifMT@if@ and a scratch if.

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

1.1.4 Declarations

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

These are the global switches . . .

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

[This line intentionally left blank.]

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

. . . and numbers.

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

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

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

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

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

Expansion settings.

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

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

Minimum and maximum values allowed by pdfTEX.

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

\MT@factor@default

Default factor.

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

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

Default values for expansion.

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

\MT@letterspace
\MT@letterspace@default

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

```
⟨/package⟩
\let\MT@letterspace\m@ne
\def\MT@letterspace@default{50}
⟨*package⟩
```

\ifMT@document

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

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

1.1.5 Auxiliary macros

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

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

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

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

```
⟨lua-⟩\IfFormatAtLeastTF{2016/01/01}\relax{\RequirePackage{luatexbase}}
```

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

```
⟨lua-⟩\IfFormatAtLeastTF{2017/01/01}\relax{\RequirePackage{luaotfload}}
⟨letterspace⟩\MT@pdf@or@lua\relax{
⟨letterspace⟩\ifx\newluafunction\@undefined \input ltluatex \fi
⟨lua-|letterspace⟩\MT@lua{require("microtype")}
⟨letterspace⟩}
```

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

```
⟨*luafile⟩

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

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

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

```

We need the function math.tointeger, which is missing in older LuaTEX versions, and ConTEXt (inherited via luaotfload) faultily overwrites its own definition. The following is the (correct) definition from l-math.lua.

```
if not math.tointeger or not pcall(math.tointeger,0) then
```

```
  math.mininteger=-0x4FFFFFFFFFFF
  math.maxinteger=0x4FFFFFFFFFFF
  local floor=math.floor
  function math.tointeger(n)
    local f=floor(n)
    return f==n and f or nil
  end
end

⟨/luafile⟩
```

To be continued, but first back to primitives.

`\MT@glet`

Here's the forgotten one (finally implemented in LuaTEX).

```
⟨lua-⟩\MT@requires@luatex6{\let\MT@glet\glet}\relax
⟨*package|letterspace⟩
\def\MT@glet{\global\let}
```

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

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

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

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

This is `\@namedef` and global.

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

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

Its expanding versions.

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

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

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

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

`\MT@let@cn`

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

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

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

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

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

`\MT@@font`

Remove trailing space from the font name.

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

`\MT@exp@one@n`

Expand the second token once and enclose it in braces.

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

`\MT@exp@two@c`

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

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

`\MT@exp@two@n`

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

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

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

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

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

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

\MT@if@expanding@F
\MT@if@expanding@F@

The following voodoo is based on a trick by *Ulrich Schwarz*.[3]

```
⟨*package⟩
\def\MT@if@expanding@F{\let\MT@if@expanding@F@\MT@if@expanding@F@\@firstofone}
\def\MT@if@expanding@F@#1#2#3{\relax\relax}
```

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

Translate a macro into a token list. With e-TEX, we can use \detokenize. We also need to remove the last trailing space; and only the last one – therefore the fiddling (and the \string isn't perfect, of course).

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

\MT@ifempty

Test whether argument is empty.

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

\MT@ifint

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

[3] Cf. https://tex.stackexchange.com/a/29188/7674

required by the letterspace option).

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

⟨/luafile⟩
```

\MT@ifdimen

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

```
⟨*pdf-⟩
\MT@requires@pdftex6{
\def\MT@ifdimen#1{%
  \ifcase\pdfmatch{^([0-9]+([.,][0-9]+)?|[.,][0-9]+)%
                    (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi
}
}{
⟨/pdf-⟩
⟨*pdf-|xe-⟩
\def\MT@ifdimen#1{%
  \setbox\z@=\hbox{%
    \MT@count=1#1\relax
    \ifnum\MT@count=\@ne
      \aftergroup\@secondoftwo
    \else
      \aftergroup\@firstoftwo
    \fi
  }%
}
⟨/pdf-|xe-⟩
⟨pdf-⟩}
```

```
⟨lua-⟩\def\MT@ifdimen#1{\csname\MT@lua{microtype.if_dimen([[#1]])}\endcsname}
⟨*luafile⟩
local function if_dimen(s)
  if (find(s, "^-*[0-9]+(%a*) *$") or
      find(s, "^-*[0-9]*[.,][0-9]+(%a*) *$")) then
    tex_write("@firstoftwo")
  else
    tex_write("@secondoftwo")
  end
end
microtype.if_dimen = if_dimen

⟨/luafile⟩
```

\MT@ifdim

Compare floating point numbers.

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

\MT@ifstreq

Test whether two strings (fully expanded) are equal.

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

⟨/luafile⟩
```

With this, we can now also check whether versions match (using the command from 1.4.2).

```
⟨lua-⟩\MT@check@MT@version
⟨lua-⟩  {\MT@lua{tex.write(microtype.module['date'] .. ' v' .. microtype.module['version'])}}
⟨lua-⟩  {\MT@MT.lua}
```

\MT@xadd

Add item to a list.

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

\MT@xaddb

Add item to the beginning.

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

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

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

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

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

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

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

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

\ifMT@inlist@
\MT@in@clist

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

```
\newif\ifMT@inlist@
\def\MT@in@clist#1#2{%
```

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

`\MT@rem@from@clist`

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

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

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

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

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

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

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

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

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

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

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

\MT@while@num

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

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

\MT@if@opentype@font

For fonts loaded by `luaotfload` we query the font's table, for XeTeX, the font type. ('opentype' here stands for non-legacy.)

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

⟨/luafile⟩
```

\MT@do@font

Execute ⟨*#1*⟩ 256 times,

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

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

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

This is the `lua` function, which is much faster than looping through all glyphs in TeX. Legacy fonts (which this function should never work on) don't contain a `v.index` field. Our test whether `i` is larger than 1114111 may seem strange, but with the HarfBuzz renderer, we are not guaranteed to get a number within the Unicode range.

```
⟨*luafile⟩
```

```
local function do_font()
  local thefont = font.getfont(font.current())
  if thefont then
    for i,v in next,thefont.characters do
      if v.index == nil or ( v.index > 0 and i < 1114112 ) then
        microtype.sprint([[\@tempcnta=]]..i..[[\relax\MT@dofont@function]])
      end
    end
  end
end
microtype.do_font = do_font

⟨/luafile⟩
```

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

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

\MT@count
\MT@increment

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

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

\MT@scale

Multiply and divide a counter. If we are using e-TEX, we will use its \numexpr primitive. This has the advantage that it is less likely to run into arithmetic overflow. The result of the division will be rounded instead of truncated. Therefore, we'll get a different (more accurate) result in about half of the cases.

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

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

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

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

```
\def\MT@abbr@kn@c{kerning codes}
\def\MT@abbr@kn@inh{kerning inheritance}
\def\MT@abbr@tr{tracking}
\def\MT@abbr@tr@c{tracking amount}
```

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

These we also need the other way round.

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

\MT@features
\MT@features@long

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

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

\MT@is@feature

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

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

1.1.6 Compatibility

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

- `\pickup@font`
- `\do@subst@correction`
- `\add@accent` (all in section 1.2.10)
- `\showhyphens` (in section 1.4.6)

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

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

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

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

\MT@setup@

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

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\let\MT@setup@\@empty
```

\MT@addto@setup

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

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

Don't hesitate with miniltx.

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

\MT@with@package@T

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

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

\MT@with@babel@and@T

LaTeX's \@ifpackagewith ignores the class options.

```
⟨*package⟩
\def\MT@with@babel@and@T#1{%
  \MT@ifdefined@n@T{opt@babel.sty}{%
    \@expandtwoargs\MT@in@clist{#1}
      {\csname opt@babel.sty\endcsname,\@classoptionslist}%
    \ifMT@inlist@\expandafter\@secondoftwo\else\expandafter\@firstofone\fi
  }\@gobble
}
⟨/package⟩
```

\MT@ledmac@setup

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

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

```
⟨*pdf-|lua-|xe-⟩
⟨pdf-⟩\MT@requires@pdftex5{
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@ifdefined@c@TF\l@dunhbox@line{%
```

\MT@led@unhbox@line

Hook.

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

      }{%
        \MT@warning@nl{%
          Character protrusion in paragraphs with line\MessageBreak
          numbering will only work if you update ledmac,\MessageBreak
          or use one of its successors, eledmac or reledmac}%
      }%
    \fi
  }
```

```
⟨*pdf-⟩
}{
  \def\MT@ledmac@setup{%
    \ifMT@protrusion
      \MT@warning@nl{%
        The pdftex version you are using does not allow\MessageBreak
        character protrusion in paragraphs with line\MessageBreak
        numbering by the `((r)e)ledmac' package.\MessageBreak
        Upgrade pdftex to version 1.30 or later}%
    \fi
  }
}
```

\MT@varwidth@setup

Likewise, the varwidth package de- and reassembles \vboxes line by line, in the course of which margin kerns will get lost. We patch the relevant commands to record and reinsert the margin kerns.

```
\MT@requires@pdftex5{
⟨/pdf-⟩
  \def\MT@varwidth@setup{%
    \ifMT@protrusion
^^Q      \MT@warning@nl{Cannot patch varwidth without etex extensions}%
^^X      \MT@info@nl{Patching varwidth to enable character protrusion}%
^^X      \newdimen\MT@vwid@leftmargin
^^X      \newdimen\MT@vwid@rightmargin
^^X      \patchcmd\@vwid@sift{\sift@deathcycles\z@}
^^X        {\MT@vwid@leftmargin\leftmarginkern\z@
^^X         \MT@vwid@rightmargin\rightmarginkern\z@ \sift@deathcycles\z@}{}{}%
^^X      \patchcmd\@vwid@resetb{\kern\@vwid@loff \unhbox\z@}
^^X        {\kern\@vwid@loff \ifdim\MT@vwid@leftmargin=\z@\else\kern\MT@vwid@leftmargin\fi
^^X         \unhbox\z@ \ifdim\MT@vwid@rightmargin=\z@\else\kern\MT@vwid@rightmargin\fi}{}{}%
^^X      \patchcmd\@vwid@measure{\kern\@vwid@loff \unhbox\z@}
^^X        {\kern\@vwid@loff \ifdim\MT@vwid@leftmargin=\z@\else\kern\MT@vwid@leftmargin\fi
^^X         \unhbox\z@ \ifdim\MT@vwid@rightmargin=\z@\else\kern\MT@vwid@rightmargin\fi}{}{}%
    \fi
  }
⟨*pdf-⟩
}{
  \def\MT@varwidth@setup{%
    \ifMT@protrusion
      \MT@warning@nl{%
        The pdftex version you are using does not allow\MessageBreak
        character protrusion in varwidth environments.\MessageBreak
        Upgrade pdftex to version 1.30 or later}%
    \fi
  }
}
⟨/pdf-⟩
⟨/pdf-|lua-|xe-⟩
```

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

\MT@restore@p@h

Restore meaning of \% and \#.

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

\ifMT@fontspec
\ifMT@xunicode

Two new conditionals for use with XƎTEX or LuaTEX.

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

We need the correct value of the former for configuration commands inside the preamble (to get the default families right).

```
\IfFormatAtLeastTF{2020/10/01}
  {\IfFormatAtLeastTF{2021/11/15}
    {\AddToHook{package/fontspec/after}{\MT@fontspectrue}}
    {\AddToHook{package/after/fontspec}{\MT@fontspectrue}}}\relax
```

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

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

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

\MT@setupfont@hook

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

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

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

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

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

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

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

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

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

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

Check again at the end of the preamble.

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

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

```
  \MT@with@package@T{pdfcprot}{%
    \MT@error{Detected the `pdfcprot' package!\MessageBreak
              `\MT@MT' and `pdfcprot' may not be used together}{%
```

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

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

```
  \MT@glet\MT@setupfont@hook\@empty
```

`microtype` is so so loquacious … Sometimes you just want to silence it when debugging a document.[4]

```
 %\gdef\MT@setupfont@hook{\ifnum\tracingmacros>\z@\tracingnone
 %  \MT@info{->Silently doing my `magic' (Mittelbach) for font\MessageBreak->\MT@@font}\fi}%
  \MT@if@false
  \MT@with@babel@and@T{spanish} \MT@if@true
  \MT@with@babel@and@T{galician}\MT@if@true
  \MT@with@babel@and@T{mexican} \MT@if@true
  \ifMT@if@
    \g@addto@macro\MT@setupfont@hook{%
      \MT@ifdefined@c@T\percentsign{\let\%\percentsign}}%
  \fi
  \MT@with@package@T{csquotes}{%
    \@ifpackagelater{csquotes}{2005/05/11}{%
      \g@addto@macro\MT@setupfont@hook\@disablequotes
```

For `\leftprotrusion`, we disable `csquotes`'s tracking of group level and type, because we'll probably be typesetting the opening quotes only.

```
      \g@addto@macro\MT@prot@hook{%
        \def\csq@bqgroup{\begingroup\leavevmode
          \let\MT@csq@eqgroup\endgroup}%
        \let\csq@eqgroup\endgroup}%
    }{%
      \MT@warning@nl{%
        Should you receive warnings about unknown slot\MessageBreak
        numbers, try upgrading the `csquotes' package}%
    }%
  }%
```

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

```
  \MT@if@false
⟨/package⟩
⟨plain⟩  \MT@requires@latex2{
  \MT@with@package@T{hyperref}{%
    \pdfstringdefDisableCommands{%
⟨*package⟩
      \MT@ltx@pickupfont
      \let\textmicrotypecontext\@secondoftwo
      \let\microtypecontext\@gobble
⟨/package⟩
      \def\lsstyle{\pdfstringdefWarn\lsstyle}%
      \def\textls#1#{\pdfstringdefWarn\textls}%
    }%
⟨package⟩    \MT@if@true
  }%
⟨plain⟩  }\relax
```

[4] Cf. https://www.youtube.com/watch?v=7FQLnggVgDE&t=38m24s

```
⟨*package⟩
  \MT@with@package@T{tex4ht}{%
    \def\MT@apply@patch#1{\MT@info{Not applying patch `#1' (for tex4ht)}}%
    \def\MT@undo@patch#1{\MT@info{Not undoing patch `#1' (for tex4ht)}}%
    \MT@if@true
  }%
  \MT@with@package@T{mathastext}\MT@if@true
  \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi
```

The `listings` package makes numbers and letters active,

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

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

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

Inside a listing, `\space` is redefined.

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

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

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

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

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

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

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

The luainputenc package makes all characters active, which can lead into problems when the unicode-math package is loaded, as the latter doesn't always define characters in LICR-conforming ways. By disabling the following command, we prevent errors; warnings about unknown slots, however, may still occur – but that's one of the unavoidable downsides of using luainputenc.

```
  \MT@with@package@T{unicode-math}{%
    \MT@let@nc{__um_sub_or_super:n}\relax
  }%
⟨/package⟩
}
⟨/package|letterspace⟩
```

1.1.7 Protrusion patches

\ifMT@patch@ok
\MT@patch@info
\MT@patch@warn
\MT@patch@undef
\MT@patch@info@undo

We have to patch some macros to get protrusion right.

```
⟨*package⟩
\newif\ifMT@patch@ok
\def\MT@patch@info#1{\MT@info{Applying patch `#1'}}
\def\MT@patch@warn#1{\MT@warning{Unable to apply patch `#1'}}
\def\MT@patch@undef#1{\MT@warning{Patch `#1' undefined.\MessageBreak Cannot apply it}}
\def\MT@patch@info@undo#1{\MT@info{Reverting patch `#1'}}
```

\MT@patches@def
\MT@define@patch

Define a patch and add it to the list of patches. The third argument may contain more revert commands, but will mostly be empty.

```
\let\MT@patches@def\@gobble
\def\MT@define@patch#1#2#3{%
  \MT@ifdefined@n@TF{MT@patch@@#1}{%
    \MT@warning{Patch `#1' already defined.\MessageBreak Cannot define it}%
  }{%
    \g@addto@macro\MT@patches@def{,#1}%
    \MT@gdef@n{MT@patch@@#1}{#2}%
    \MT@gdef@n{MT@patch@undo@@#1}{#3}%
  }%
}
```

\MT@redefined@patches
\MT@redefine@patch

We also provide an easier way of redefining patches, which would otherwise be a bit tricky because of the timing (patches are defined *and* executed ABD).

```
\let\MT@redefined@patches\@empty
\def\MT@redefine@patch#1#2#3{%
  \g@addto@macro\MT@redefined@patches{%
    \MT@ifdefined@n@TF{MT@patch@@#1}{%
      \MT@gdef@n{MT@patch@@#1}{#2}%
      \MT@gdef@n{MT@patch@undo@@#1}{#3}%
    }{%
      \MT@warning{Patch `#1' undefined.\MessageBreak Cannot redefine it}%
    }%
  }%
}
```

Both macros are only allowed in the preamble.

```
\@onlypreamble\MT@define@patch
\@onlypreamble\MT@redefine@patch
```

\MT@append@patch
\MT@patch@patch

Wrappers around etoolbox commands. We also remember the original command to allow unpatching.

```
\def\MT@append@patch#1#2{%
  \MT@remember@patch{#1}%
  \apptocmd#1{#2}\relax\MT@patch@okfalse
}
```

```
\def\MT@patch@patch#1#2#3{%
  \MT@remember@patch{#1}%
  \patchcmd#1{#2}{#3}\relax\MT@patch@okfalse
}
```

\MT@remember@patch

Remember the original definition and add to undo command.

```
\def\MT@remember@patch#1{%
  \MT@ifdefined@n@TF{MT@patch@saved@\string#1}\relax
    {\MT@let@nc{MT@patch@saved@\string#1}#1%
     \MT@exp@cs\g@addto@macro{MT@patch@undo@@\MT@patch@name}%
       {\MT@let@cn#1{MT@patch@saved@\string#1}}}%
}
```

\MT@patches@applied
\MT@apply@patch

Apply a previously defined patch. With some packages, we have to reset catcodes (e.g., for the '`item`' patch with Spanish `babel`, which makes '>' active).

```
\let\MT@patches@applied\@gobble
\def\MT@apply@patch#1{%
  \MT@patch@oktrue
  \MT@ifdefined@n@TF{MT@patch@@#1}
    {\MT@in@clist{#1}\MT@patches@applied
     \ifMT@inlist@
       \MT@warning{Patch `#1' has already been applied,\MessageBreak
                   cannot reapply it}%
     \else
       \let\MT@restore@catcodes\@empty
       \MT@with@babel@and@T{spanish} {\MT@fix@catcode{62}{12}}% >
       \MT@with@babel@and@T{galician}{\MT@fix@catcode{62}{12}}% >
       \def\MT@patch@name{#1}%
       \g@addto@macro\MT@patches@applied{,#1}%
       \@nameuse{MT@patch@@#1}%
       \@nameuse{MT@patch@\ifMT@patch@ok info\else warn\fi}{#1}%
       \MT@restore@catcodes
     \fi}
    {\MT@patch@undef{#1}}%
}
```

\MT@undo@patch

Undo a patch (if indeed previously applied).

```
\def\MT@undo@patch#1{%
  \MT@in@clist{#1}\MT@patches@applied
  \ifMT@inlist@
    \MT@rem@from@clist{#1}\MT@patches@applied
    \@nameuse{MT@patch@undo@@#1}%
    \MT@patch@info@undo{#1}%
  \else
    \MT@warning{Patch `#1' hasn't been applied,\MessageBreak cannot revert it}%
  \fi
}
```

Unfortunately, `etoolbox` is a bit bitchy with hashes in arguments (but who would blame it), so I currently see no other solution than to temporarily reset the catcode of the # character.

```
{\catcode`\#=12
\MT@addto@setup{%
```

Now for the actual patches:

`item`: `\@item`, which is a kind of catch-all, as it's internally used for most basic environments (e.g., `itemize`, `enumerate`, but also `quote`, `flushleft` etc.). For `verse` (and probably other environments), we also have to patch `\everypar` ...

- for the base classes

```
      \MT@define@patch{item}{%
        \MT@append@patch\@item\leftprotrusion
```

```
    \MT@patch@patch\@item{\everypar{}}{\everypar{\leftprotrusion}}%
```

- beamer patches it too

```
    \@ifclassloaded{beamer}
      {\MT@append@patch\beamer@@callorigitem\leftprotrusion
       \MT@patch@patch\beamer@callorigitem{\ignorespaces}{\ignorespaces\leftprotrusion}}
```

- the simplecv class

```
      {\@ifclassloaded{simplecv}
        {\MT@append@patch\@topic@item\leftprotrusion}
        {}}%
  }{}%
```

toc: TOC and friends

```
  \MT@define@patch{toc}{%
    \MT@append@patch\numberline\leftprotrusion
```

- for the memoir class we also fix the extra leader problem . . .

```
    \@ifclassloaded{memoir}
      {\MT@append@patch\booknumberline\leftprotrusion
       \MT@append@patch\partnumberline\leftprotrusion
       \MT@append@patch\chapternumberline\leftprotrusion
       \MT@append@patch\cftbookafterpnum\noprotrusion
       \MT@append@patch\cftpartafterpnum\noprotrusion
       \MT@append@patch\cftchapterafterpnum\noprotrusion
       \MT@append@patch\cftsectionafterpnum\noprotrusion
       \MT@append@patch\cftsubsectionafterpnum\noprotrusion
       \MT@append@patch\cftsubsubsectionafterpnum\noprotrusion
       \MT@append@patch\cftparagraphafterpnum\noprotrusion
       \MT@append@patch\cftsubparagraphafterpnum\noprotrusion
       \MT@append@patch\cftfigureafterpnum\noprotrusion
       \MT@append@patch\cfttableafterpnum\noprotrusion}
      {}%
  }{}%
```

- for the KOMA classes (which load the tocbasic package) we additionally have to switch protrusion back on; this will re-introduce the risk of getting an extra leader dot, but I currently don't see how to easily add \noprotrusion. Therefore, I'll skip this patch for now, saving the joy of wading through scr files for later, all the while waiting for somebody who would understand KOMA better than me.

```
%    \@ifpackageloaded{tocbasic}
%      {\MT@define@patch{toc}
%        {\MT@append@patch\numberline\leftprotrusion
%         \setuptoc{toc}{noprotrusion}%
%         \setuptoc{lof}{noprotrusion}%
%         \setuptoc{lot}{noprotrusion}}
%        {\unsettoc{toc}{noprotrusion}%
%         \unsettoc{lof}{noprotrusion}%
%         \unsettoc{lot}{noprotrusion}}}{}%
```

- (a patch for titletoc would also be worthwhile . . .)

eqnum: equation numbers

- IEEEtran

```
  \MT@define@patch{eqnum}{%
    \@ifclassloaded{IEEEtran}
      {\MT@patch@patch\theequationdis{(}{\leftprotrusion{(}}%
       \MT@patch@patch\theequationdis{)}{\rightprotrusion{)}}%
```

```
     \MT@patch@patch\theIEEEsubequationdis{(}{\leftprotrusion{(}}%
     \MT@patch@patch\theIEEEsubequationdis{)}{\rightprotrusion{)}}}%
    {}%
```

- \eqref (amsmath) relies on \tagform@, so we have to have it use the original definition. The showkeys package also modifies this command, as well as \@eqnnum (below); we don't test for the package itself but the relevant command, which is only redefined ABD.

```
  \@ifpackageloaded{amsmath}
    {\MT@ifdefined@c@TF\SK@tagform@
      {\MT@patch@patch\SK@tagform@{(}{\leftprotrusion{(}}%
       \MT@patch@patch\SK@tagform@{)}{\rightprotrusion{)}}}%
      {\MT@patch@patch\tagform@{(}{\leftprotrusion{(}}%
       \MT@patch@patch\tagform@{)}{\rightprotrusion{)}}}%
```

The command has been made robust in 2022.

```
     \MT@ifdefined@n@TF{eqref }
       {\MT@exp@cs\MT@patch@patch{eqref }}{\MT@patch@patch\eqref}
         {\tagform@}{\@nameuse{MT@patch@saved@\string\tagform@}}%
```

- If the user has altered the tags' appearance via mathtools's \newtagform interface, our patch won't have any effect. We don't issue a warning because \(left|right)protrusion might have been specified appropriately in \newtagform. We could also patch the latter command (or, to be more precise, \MT_define_tagform:nwnn), but the timing is a bit tricky, so for now info it is.

```
     \MT@with@package@T{mathtools}{%
       \ifMT@patch@ok\else \MT@patch@oktrue
         \MT@info@nl{The `eqnum' patch may not be effective because you are\MessageBreak
             using the mathtools package. Make sure to insert\MessageBreak
             `\@backslashchar leftprotrusion' and
             `\@backslashchar rightprotrusion' as\MessageBreak
             appropriate in mathtools's `\@backslashchar newtagform' command}%
       \fi}}
    {\@ifclassloaded{IEEEtran}{}
      {\MT@ifdefined@c@TF\SK@eqnnum
        {\MT@patch@patch\SK@eqnnum{(}{\leftprotrusion{(}}%
         \MT@patch@patch\SK@eqnnum{)}{\rightprotrusion{)}}}%
        {\MT@patch@patch\@eqnnum{(}{\leftprotrusion{(}}%
         \MT@patch@patch\@eqnnum{)}{\rightprotrusion{)}}}}}%
}{}%
```

footnote: footnote text (only visible with block paragraphs)

- The new footnote code (in latex-lab-footnotes.ltx), which is meant to facilitate tagging, introduces many hooks, among them fntext/begin, which would seem appropriate for us. Unfortunately, however, we cannot use it, as we'd stumble over the hook management itself. I think it's unnecessary to patch the expl3 version too, but I suppose it won't do any harm either. There's no new code for footnotes in minipages at the moment.

```
\MT@define@patch{footnote}{%
  \MT@ifdefined@n@TF{fnote_footnotetext:n}
    {\ExplSyntaxOn
     \MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
     \MT@exp@cs\MT@patch@patch{fnote_footnotetext:n}{\ignorespaces}{\ignorespaces\leftprotrusion}%
     \MT@patch@patch\@mpfootnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
     \ExplSyntaxOff}
```

- hyperref also patches this command (but only if hyperfootnotes=true, implicit=true and \hyper@nopatch@footnote is undefined)

```
          {\@ifpackageloaded{hyperref}
            {\MT@if@false
             \ifHy@implicit
               \ifHy@hyperfootnotes
                 \MT@ifdefined@c@TF\hyper@nopatch@footnote\relax
                   \MT@if@true
               \fi
             \fi
             \ifMT@if@\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
            \@secondoftwo
          {\MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
           \MT@patch@patch\@footnotetext{\@empty\ignorespaces}{\@empty\ignorespaces\leftprotrusion}%
           \MT@patch@patch\@mpfootnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}%
           \MT@patch@patch\@mpfootnotetext
               {{\expandafter\hyper@@anchor\expandafter
                   {\Hy@footnote@currentHref}{\relax}}\ignorespaces}
               {{\expandafter\hyper@@anchor\expandafter
                   {\Hy@footnote@currentHref}{\relax}}\ignorespaces\leftprotrusion}}
```

- memoir additionally allows footnotes in the margins

```
          {\@ifclassloaded{memoir}
            {\MT@patch@patch\@footnotetext{\foottextfont #1}{\foottextfont\leftprotrusion #1}%
             \MT@patch@patch\@mpfootnotetext{\foottextfont #1}{\foottextfont\leftprotrusion #1}}
```

- beamer has it its own way, of course

```
            {\@ifclassloaded{beamer}
              {\MT@exp@cs\MT@patch@patch{beamerx@\string\beamer@framefootnotetext}
                   {\ignorespaces}{\ignorespaces\leftprotrusion}%
               \MT@exp@cs\MT@patch@patch{beamerx@\string\@mpfootnotetext}
                   {\ignorespaces}{\ignorespaces\leftprotrusion}}
```

- the KOMA classes

```
              {\MT@ifdefined@c@TF\KOMAClassName
                {\MT@patch@patch\scr@saved@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}%
```

- the base classes

```
                {\MT@patch@patch\@footnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}%
               \MT@patch@patch\@mpfootnotetext{\ignorespaces}{\ignorespaces\leftprotrusion}}}}}%
      }{}%
```

verbatim: disable all microtypographic extensions in verbatim blocks. (This could have been another nice opportunity to use the new LaTeX hook management, however, the hook here is executed too early – namely, before the \par in \@verbatim, which may result in spilling the microtypographic settings to the preceding paragraph – so we're resorting to patching, again.)

- Appending to \@verbatim works for, at least, the standard classes, verbatim (and memoir); the implementations in fancvrb and listings don't allow protrusion anyway.

```
      \MT@define@patch{verbatim}{%
        \MT@append@patch\@verbatim{\microtypesetup{activate=false}}%
```

- package alltt

```
        \MT@with@package@T{alltt}{\MT@append@patch\alltt{\microtypesetup{activate=false}}}%
      }{}
```

Finally, execute any redefinitions.

```
  \MT@redefined@patches
}}
⟨/package⟩
```

1.2 Font setup

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

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

\MT@setupfont

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

```
⟨*pdf-|lua-|xe-⟩
\def\MT@setupfont{%
```

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

```
⟨xe-|lua-⟩  \MT@font
```

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

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

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

```
⟨pdf-⟩\MT@requires@pdftex7{
⟨pdf-|lua-⟩\g@addto@macro\MT@setupfont\MT@copy@font
⟨pdf-⟩}\relax
```

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

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

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

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

We have to make sure that `\cf@encoding` expands to the correct value (for later, in `\MT@get@slot`), which isn't the case when `\selectfont` chooses a new encoding (this would be done a second later in `\selectfont`, anyway – three lines, to be exact). (I think, I do not need this anymore – however, I'm too afraid to remove it. . . . Oops, I did it. Let's see whether anybody complains.)

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

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

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

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

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

```
  \MT@protrusion
```

```
⟨pdf-|lua-⟩  \MT@expansion
}
```

Interword spacing and kerning (pdfTEX 1.40).

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

Disable ligatures (pdfTEX 1.30).

```
⟨pdf-⟩\MT@requires@pdftex5{
⟨pdf-|lua-⟩\g@addto@macro\MT@setupfont\MT@noligatures
⟨pdf-⟩}\relax
\g@addto@macro\MT@setupfont{%
```

Debugging.

```
⟨debug⟩\MT@show@pdfannot1%
```

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

```
    \MT@register@font
  \fi
}
⟨/pdf-|lua-|xe-⟩
```

\MT@copy@font
\MT@copy@font@

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

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

\MT@font@copy

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

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

\MT@font@orig

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

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

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

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

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

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

```
  \fi
  \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

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

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

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

\MT@rem@from@list

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

\MT@lua@copy@font

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

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

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

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

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

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

If `\fontdimen` 6 is zero, character protrusion, spacing, kerning and tracking won't work, and we could skip the settings (for example, the `dsfont` fonts don't specify this dimension; this is probably a bug – the `fourier` and `newpx`/`newtx` packages have been fixed in the meantime). However, we can fix it ourselves (and since pdfTEX 1.40.23, this also works for `\letterspacefont`). XƎTEX (and newer LuaTEX in DVI mode) doesn't provide an equivalent to `\pdffontsize`, so we use the nominal size instead.

```
⟨*pdf-|lua-|xe-⟩
\def\MT@fix@fontdimen@six{%
  \ifnum\fontdimen6\MT@font=\z@
    \fontdimen6\MT@font=%
⟨pdf-⟩      \pdffontsize\MT@font
⟨lua-⟩      \MT@requires@luatex4{\ifnum\outputmode=\@ne \pdffeedback fontsize\else
⟨lua-|xe-⟩              \MT@size pt%
⟨lua-⟩                          \expandafter\@gobble\fi}{\pdffontsize}\MT@font
    \MT@info{Fixing zero \@backslashchar fontdimen 6 for font `\MT@@font'\MessageBreak
             (new value: \the\fontdimen6\MT@font)}%
⟨pdf-⟩    \MT@requires@pdftex8\relax{\MT@glet@nc{\MT@@font-fake6}\@empty}%
  \fi
  \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
}
⟨/pdf-|lua-|xe-⟩
```

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

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

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

\MT@familyalias

Alias family?

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

\MT@scrubfeature \MT@scrubfeatures

Remove one resp. all feature counters (fontspec).

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

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

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

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

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

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

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

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

```
    \MT@dofalse
  \fi
  \ifMT@do
```

\MT@feat stores the current feature.

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

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

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

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

\MT@dinfo@list

```
⟨debug⟩\def\MT@dinfo@list#1#2#3{\MT@dinfo@nl{1}{\@nameuse{MT@abbr@#1}: #2
⟨debug⟩  \ifx\\#3\\list empty\else `\@nameuse{MT@#2}' #3 list\fi}}
```

\MT@checklist@

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

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

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

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

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

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

\MT@checklist@family

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

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

\MT@checklist@size

Test whether font size is in list of size ranges.

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

\MT@checklist@font

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

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

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

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

1.2.1 Protrusion

\ifMT@nofamily

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

```
\newif\ifMT@nofamily
```

\MT@protrusion

Set up for protrusion?

```
\def\MT@protrusion{\MT@maybe@do{pr}}
⟨/package⟩
```

\MT@set@pr@codes

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

```
⟨*pdf-|lua-|xe-|show⟩
⟨show⟩\def\MTS@show@pr
⟨pdf-|lua-|xe-⟩\def\MT@set@pr@codes
  {%
⟨pdf-|lua-|xe-⟩ \MT@nofamilyfalse
```

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

```
⟨show⟩  \MTS@printtext{Protrusion settings for font `\texttt{\MT@@font}':}\\
  \MT@if@list@exists{%
⟨*pdf-|lua-|xe-⟩
    \ifMT@nofamily
      \MT@ifdefined@n@TF{\MT@encoding-\MT@family-settings}\relax{%
        \MT@info@nl{Loading generic protrusion settings for font family\MessageBreak
                    `\MT@family' (encoding: \MT@encoding).\MessageBreak
                    For optimal results, create family-specific settings.\MessageBreak
                    See the microtype manual for details}%
        \MT@glet@nc{\MT@encoding-\MT@family-settings}\@empty
      }%
    \fi
⟨/pdf-|lua-|xe-⟩
⟨show⟩    \MTS@printtext{First matching list is for `\texttt{\@tempa}':\\\texttt{\MT@pr@c@name}}%
    \MT@get@opt
    \MT@reset@pr@codes
```

Get the name of the inheritance list and parse it.

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

Set an input encoding?

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

Load additional lists?

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

Load the main list.

```
    \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
    \expandafter\MT@set@codes\@tempc,\relax,%
⟨show⟩    \vrule width 4cm height .5pt \\
⟨show⟩    \MTS@printtext{End of list `\texttt{\MT@pr@c@name}'}\\[.5em]
⟨show⟩    \MT@ifdefined@c@T\MT@pr@inh@name{%
⟨show⟩      \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @prefixes}{%
⟨show⟩        \par \MTS@printtext{(with prefixes:)}%
⟨show⟩        \@tempcntb=\z@
```

Set unconditional heirs.

```
    \MT@set@pr@prefixheirs
⟨show⟩    }}%
⟨show⟩    \ifShowMissingGlyphs\MTS@show@missing\fi
  }%
⟨show⟩  {\MTS@printtext{NOT DEFINED}%
  \MT@reset@pr@codes
⟨show⟩  }\par
}
```

\MT@set@all@pr

Set all protrusion codes of the font.

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

\MT@reset@pr@codes@
\MT@reset@pr@codes

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

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

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

If the font is letterspaced, we have to add half the letterspacing amount to the margin kerns. This will be activated in \MT@set@tr@codes.

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

\MT@set@codes

Split up the values and set the codes.

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

\MT@split@codes

The keyval package would remove spaces here, which we needn't do since \SetProtrusion ignores spaces in the protrusion list anyway. \MT@get@char@unit may mean different things.

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

\MT@pr@split@val

```
\def\MT@pr@split@val#1,#2\relax
⟨/pdf-|lua-|xe-⟩
⟨show⟩\def\MTS@pr@split@val#1,#2\relax
  {\def\@tempb{#1}%
  \MT@ifempty\@tempb
⟨pdf-|lua-|xe-⟩     \relax
⟨show⟩  {\MTS@lp@=\z@ \let\MTS@lpcode\@empty}%
  {\MT@scale@to@em
⟨pdf-|lua-|xe-⟩     \lpcode\MT@font\MT@char=\MT@the@pr@code
⟨show⟩     \MTS@lp@=\dimexpr\@tempcntb em/1000\relax\relax
⟨show⟩     \edef\MTS@lpcode{[\@tempb] \the\@tempcntb/\the\MTS@lp@}%
⟨debug⟩\MT@dinfo@nl{4}{;;; lp (\MT@char): \number\lpcode\MT@font\MT@char: [#1]}%
  }%
  \def\@tempb{#2}%
  \MT@ifempty\@tempb
⟨pdf-|lua-|xe-⟩     \relax
⟨show⟩  {\MTS@rp@=\z@ \let\MTS@rpcode\@empty}%
  {\MT@scale@to@em
```

```
⟨pdf-|lua-|xe-⟩     \rpcode\MT@font\MT@char=\MT@the@pr@code
⟨show⟩     \MTS@rp@=\dimexpr\@tempcntb em/1000\relax\relax
⟨show⟩     \edef\MTS@rpcode{[\@tempb] \the\@tempcntb/\the\MTS@rp@}%
⟨debug⟩\MT@dinfo@nl{4}{;;; rp (\MT@char): \number\rpcode\MT@font\MT@char: [#2]}%
  }%
⟨show⟩  \llap{\MTS@show@char@pr\MT@char\quad}%
⟨show⟩  \parbox[b][][b]{3.5cm}{\MTS@printtext{%
⟨show⟩      \footnotesize\makebox[.4cm][l]{L:} \MT@ifempty{\MTS@lpcode}{---}{\MTS@lpcode}\\
⟨show⟩                   \makebox[.4cm][l]{R:} \MT@ifempty{\MTS@rpcode}{---}{\MTS@rpcode}}}%
⟨show⟩  \parbox[t][][t]{\dimexpr\textwidth-3.5cm}{%
```

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 @}%
⟨pdf-|lua-|xe-⟩        \MT@set@pr@heirs
⟨show⟩      \MTS@show@char@pr
    }%
  }%
⟨show⟩  }\newline
}
⟨*pdf-|lua-|xe-⟩
```

\MT@scale@to@em

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

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

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

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

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

\MT@get@charwd

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

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

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

```
⟨*xe-⟩
```

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

For letterspaced fonts, we have to subtract the letterspacing amount from the characters' widths. The protrusion amounts will be adjusted in \MT@set@pr@codes. The letterspaced font is already loaded so that 1 em = \fontdimen 6.

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

No adjustment with versions 0.14f and 0.14g.

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

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

```
\def\MT@get@charwd{\MT@count=\MT@dimen@six}
}
⟨/pdf-⟩
⟨/pdf-|lua-|xe-⟩
⟨/pdf-|lua-|xe-|show⟩
```

\MT@get@font@dimen

For the space unit.

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

\MT@info@missing@char

Info about missing characters, or characters with zero width.

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

\MT@scale@factor

Furthermore, we might have to multiply with a factor.

```
\def\MT@scale@factor{%
  \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
```

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

\MT@warn@code@too@large

Type out a warning if a chosen protrusion factor is too large after the conversion. As a special service, we also type out the maximum amount that may be specified in the configuration.

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

\MT@get@opt

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

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

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

Apply a factor?

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

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

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

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

\MT@get@space@unit
\MT@get@char@unit

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

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

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

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

\MT@get@unit
\MT@get@unit@

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

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

\MT@set@inputenc

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

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

\MT@cat

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

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

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

\MT@set@inputenc@

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

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

\MT@load@inputenc

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

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

\MT@set@pr@heirs

Set the inheriting characters.

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

\MT@set@pr@prefixheirs

Inheriting characters that have been specified in a prefixed list.

```
\def\MT@set@pr@prefixheirs{%
  \MT@ifdefined@c@T\MT@pr@inh@name{%
    \MT@ifdefined@n@T{MT@inh@\MT@pr@inh@name @prefixes}{%
      \MT@exp@cs\MT@map@tlist@c
        {MT@inh@\MT@pr@inh@name @prefixes}%
        \MT@set@pr@prefixes
    }%
  }%
}
⟨/package⟩
```

\MT@set@pr@prefixes
\MT@set@pr@prefixes@

Add `charwidth(`⟨*inheriting char*⟩`)-charwidth(`⟨*base char*⟩`)` to either left or right side or half the amount to both sides. For XƎTEX, we may have to translate to glyph numbers because `\fontcharwd` doesn't have the nice feature of understanding the '`U`' or '`/`' prefixes.

```
⟨*pdf-|lua-|xe-|show⟩
⟨pdf-|lua-|xe-⟩\def\MT@set@pr@prefixes#1{\MT@set@pr@prefixes@#1}
⟨pdf-|lua-|xe-⟩\def\MT@set@pr@prefixes@#1#2#3#4%
⟨show⟩\def\MTS@set@pr@prefixes@#1#2#3#4%
  {%
⟨show⟩  \MTS@lp@=\z@ \MTS@rp@=\z@
⟨show⟩  \ifnum#1=\@tempcntb \else
⟨show⟩    \par\leavevmode
⟨show⟩    \llap{\MTS@show@char@pr{#1} \MTS@printtext{=} }%
⟨show⟩  \fi
⟨*xe-⟩
  \edef\@tempa{\expandafter\ifx\@car#1\@nil U\@gobble#1\else\number\XeTeXglyphindex"#1" \fi}%
  \edef\@tempb{\expandafter\ifx\@car#2\@nil U\@gobble#2\else\number\XeTeXglyphindex"#2" \fi}%
⟨/xe-⟩
  \@tempcnta=\z@
  \ifnum#3>\z@
    \@tempcnta=\numexpr
⟨pdf-|lua-|show⟩      (\fontcharwd\MT@font#2-\fontcharwd\MT@font#1)%
⟨xe-⟩      (\fontcharwd\MT@font\@tempb-\fontcharwd\MT@font\@tempa)%
      *#3/\MT@dimen@six\relax
  \fi
⟨pdf-|lua-|xe-⟩  \lpcode\MT@font #2=\numexpr\lpcode\MT@font#1+\@tempcnta\relax
⟨show⟩  \MTS@lp@=\dimexpr\numexpr\lpcode\MT@font#1+\@tempcnta\relax em/1000\relax
  \@tempcnta=\z@
  \ifnum#4>\z@
    \@tempcnta=\numexpr
⟨pdf-|lua-|show⟩      (\fontcharwd\MT@font#2-\fontcharwd\MT@font#1)%
⟨xe-⟩      (\fontcharwd\MT@font\@tempb-\fontcharwd\MT@font\@tempa)%
      *#4/\MT@dimen@six\relax
  \fi
⟨pdf-|lua-|xe-⟩  \rpcode\MT@font #2=\numexpr\rpcode\MT@font#1+\@tempcnta\relax
⟨show⟩  \MTS@rp@=\dimexpr\numexpr\rpcode\MT@font#1+\@tempcnta\relax em/1000\relax
⟨debug⟩\MT@dinfo@nl{2}{-- (prefix) heir of #1: #2}%
⟨debug⟩\MT@dinfo@nl{4}{;;; lp/rp (#2): \number\lpcode\MT@font#2/%
⟨debug⟩                                  \number\rpcode\MT@font#2}%
⟨show⟩  \MTS@show@char@pr{#2}%
⟨show⟩  \@tempcntb=#1\relax
}
⟨/pdf-|lua-|xe-|show⟩
```

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

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

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

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

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

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

\MT@warn@preset@towidth

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

1.2.2 Manual protrusion

\noprotrusion

This command may be used to inhibit protrusion on either side. It's part of LaTeX since 2018-12-01. We provide it for older releases.

```
\MT@ifdefined@c@TF\noprotrusion\relax{
  \DeclareRobustCommand\noprotrusion{\leavevmode\kern-\p@\kern\p@}
}
```

\noprotrusionifhmode

Same, but only if we're already in hmode.

```
\DeclareRobustCommand\noprotrusionifhmode{\relax\ifhmode\kern-\p@\kern\p@\fi}
```

\leftprotrusion

This command may be used to add protrusion on the left hand side. We try to reconstruct the next glyph (possibly a ligature).[5]

```
\DeclareRobustCommand\leftprotrusion{%
  \MT@toks{}%
  \MT@prot@toks{}%
  \let\MT@prot@l\MT@prot@l@
  \let\MT@prot@get@first@group\MT@prot@get@first@group@
  \let\MT@maybe@textcmd\@firstofone
  \MT@prot@get@firstgroup
}
```

\MT@prot@l

This probably doesn't need to be \long any longer.

\MT@prot@l@

```
\def\MT@prot@l@#1{%
  \MT@get@prot{#1}{left}%
  #1%
}
```

\MT@prot@toks
\MT@prot@l@tc
\MT@gobble@to@nil

If \leftprotrusion is followed by a text command, we trial-typeset only the first glyph, then actually typeset the whole argument, which we've saved in \MT@prot@toks, and finally gobble anything that might still be left in the input stream (see \MT@prot@check@F below).

```
\newtoks\MT@prot@toks
\def\MT@prot@l@tc#1{%
  \MT@get@prot{\MT@maybe@textcmd{#1}}{left}%
  \the\MT@prot@toks
  \MT@gobble@to@nil
}
\def\MT@gobble@to@nil#1\MT@nil{}
```

\rightprotrusion
\MT@prot@r

Unfortunately, there's no way to retrieve anything that's already been typeset, so the counterpart cannot be defined symmetrically.

```
\DeclareRobustCommand\rightprotrusion{\MT@prot@r}
\def\MT@prot@r#1{%
  {#1}%
  \MT@get@prot{#1}{right}%
}
```

\MT@get@prot

Typeset the text inside a box and get the left and right margin kerns. We add an extra \vbox in case we're inside a tabular. \@newlistfalse is meant to make \\ work in centering etc. We set various penalties to zero to allow linebreaking, and don't bother if the split box is overfull (but shouldn't we? – after all, that's how the

[5] LuaTeX offers the command \protrusionboundary, which could potentially be very helpful here, but it doesn't seem to do what it promises (not even the example from the manual works as advertised). Maybe *Marcel Krüger*'s attempt at a betterprotrusionboundary (https://tex.stackexchange.com/a/629080) could be an option.

penalties bug was discovered . . .). (We no longer reset counters etc., since we don't typeset whole arguments anymore.) Also, we begin a group to make it color-safe.

\MT@prot@hook

Furthermore, we have a hook for compatibility fixes (currently used for csquotes only),

\MT@csq@eqgroup

and a dedicated command to end csquotes's group (because we actually typeset the quote character, instead of disabling quotes altogether (as we suggested for [issue #1], which was wrong)). Compatibility with csquotes is also the reason for the extra \relax after ⟨*#1*⟩.

\MT@noindent

Finally, LaTeX's new paragraph hooks require special attention, as they're (currently?) unable to distinguish between real typesetting and trial runs. In our case, fortunately, we really don't want to trigger the hooks.[6] Also, as far as I can tell, we don't need a \RawParEnd at the end (as suggested in ltpara), because none of our commands are \long anymore.

```
\let\MT@prot@hook\@empty
\let\MT@csq@eqgroup\relax
\IfFormatAtLeastTF{2021/11/15}
  {\let\MT@noindent\RawNoindent}
  {\let\MT@noindent\noindent}
\def\MT@get@prot#1#2{%
  \begingroup
    \setbox\MT@tempbox\vbox{%
      \everypar{}%
      \parfillskip=\z@skip
      \hbadness\@M
      \clubpenalty\z@
      \widowpenalty\z@
      \interlinepenalty\z@
      \@newlistfalse
      \MT@prot@hook
      \begingroup
        \MT@noindent #1\relax\MT@csq@eqgroup
      \endgroup}%
    \vbadness=\@M
    \splittopskip=\z@
    \vfuzz=\maxdimen
    \setbox\MT@tempbox\vbox{%
      \ifvbox\MT@tempbox
        \global\setbox\MT@tempbox=\vsplit\MT@tempbox to \normalbaselineskip
        \unvbox\MT@tempbox
        \global\setbox\MT@tempbox=\lastbox
      \fi
    }%
  \endgroup
  \ifhbox\MT@tempbox
    \@tempdima=\@nameuse{#2marginkern}\MT@tempbox\relax
    \expandafter\ifdim\@tempdima=\z@ \else
      \leavevmode
      \MT@vinfo{|<< adding #2 margin kern for `#1':\MessageBreak
        \the\@tempdima \on@line}%
      \kern\@tempdima
⟨debug⟩%\vbox to0pt{\vss\llap{\fbox{%
⟨debug⟩%       \MT@ifstreq{#2}{left}{\kern\@tempdima}\relax
⟨debug⟩%       \kern-\fboxsep\unhbox\MT@tempbox\kern-\fboxsep
⟨debug⟩%       \MT@ifstreq{#2}{right}{\kern\@tempdima}\relax}\hskip\marginparsep}}%
    \fi
  \fi
}
```

[6] Well, in some cases we do, but this indeed 'needs further analysis' (cf. https://github.com/latex3/latex2e/issues/880).

\MT@prot@ifx

Test next token.

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

\MT@prot@ifcat

Test catcode of next token.

```
\def\MT@prot@ifcat#1{%
  \ifcat#1\noexpand\MT@prot@next\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
}
```

\MT@prot@ifmacro
\MT@prot@ifmacro@

Test whether ⟨*#1*⟩ is a macro or an active character that does not take an argument. As we're using etoolbox here, this only works with e-TEX.

```
^^X\def\MT@prot@ifmacro@{%
^^X  \ifdefmacro\MT@prot@next{\ifdefparam\MT@prot@next\@gobble\@firstofone}\@gobble}
^^Q\let\MT@prot@ifmacro\@gobble
```

\MT@prot@iffirstcmd

Test whether the first token in \MT@prot@next (once expanded) is the command ⟨*#1*⟩. Since \MT@prot@next may also be user-defined (or whatever), we have to use our own, \long version of \@car.

```
\def\MT@prot@iffirstcmd#1{%
  \ifx\relax#1\expandafter\@secondoftwo\else
    \MT@exp@two@c\ifx\MT@car\MT@prot@next\relax\@nil#1%
      \expandafter\expandafter\expandafter\@firstoftwo
    \else
      \expandafter\expandafter\expandafter\@secondoftwo
    \fi
  \fi
}
```

\MT@car

A long car.

```
\long\def\MT@car#1#2\@nil{#1}
```

\MT@prot@iflicrcmd

Fun with LICR: If we have an encoding command, test if the first command of the third command (e.g., \T1\") is \@text@composite, in which case also grab the next token, otherwise it should be a text command.

```
\def\MT@getthird#1#2#3#4\@nil{#3}
\def\MT@prot@iflicrcmd{%
  \MT@prot@iffirstcmd\@current@cmd\@secondoftwo\@firstofone
    {\MT@prot@iffirstcmd\@changed@cmd\@firstofone\@gobble}%
  {\expandafter\expandafter\expandafter\let
      \expandafter\expandafter\expandafter\@tempa
      \expandafter\MT@getthird\MT@prot@next\relax\@nil
    \MT@exp@two@c\ifx\@car\@tempa\relax\@nil\@text@composite
      \def\MT@temp*##1##2{\MT@exp@one@n\MT@prot@l{\the\MT@toks##1##2}}%
    \else
      \def\MT@temp*##1{\MT@exp@one@n\MT@prot@l{\the\MT@toks##1}}%
    \fi
  }%
}
```

\MT@prot@addgroup

If we have a group, we inject \MT@prot@get@firstgroup again at the beginning and don't bother about the rest. This still allows, e.g., \verb, verbatim or lstlistings material. The downside of being this cautious is that we'll miss lots of cases.

```
\def\MT@prot@addgroup{\bgroup\afterassignment\MT@prot@get@firstgroup\let\MT@temp= }
```

\MT@prot@get@firstgroup
\MT@prot@get@firstgroup@tc
\MT@prot@get@firsttoken
\MT@prot@get@nexttoken

Scan token by token.

```
\def\MT@prot@get@firstgroup{\futurelet\MT@prot@next\MT@prot@get@first@group}
\def\MT@prot@get@firstgroup@tc{\futurelet\MT@prot@next\MT@prot@get@first@group@tc}
\def\MT@prot@get@firsttoken{\futurelet\MT@prot@next\MT@prot@get@first@token}
\def\MT@prot@get@nexttoken{\futurelet\MT@prot@next\MT@prot@get@next@token}
```

\MT@prot@get@first@group
\MT@prot@get@first@group@

If next char is {, start a group and try again, else continue until we find a beginning char.

```
\def\MT@prot@get@first@group@{%
  \MT@prot@ifcat\bgroup{%
    \def\MT@temp*{\MT@prot@addgroup}%
  }{%
    \def\MT@temp*{\MT@prot@get@first@token}%
  }%
  \MT@temp*%
}
```

\MT@prot@get@first@group@tc

The variant for text commands (in case they start with another group).

```
\def\MT@prot@get@first@group@tc{%
  \MT@prot@ifcat\bgroup{%
    \def\MT@temp*##1##2\MT@nil{\MT@ifempty{##1}\relax
          {{\MT@prot@get@firstgroup@tc##1\MT@nil}}}%
  }{%
    \def\MT@temp*{\MT@prot@get@first@token}%
  }%
  \MT@temp*%
}
```

\MT@prot@get@first@token

This can be called repeatedly. We add a letter or other character, . . .

```
\def\MT@prot@get@first@token{%
  \def\MT@temp*{\MT@exp@one@n\MT@ifempty{\the\MT@toks}
      {\MT@exp@one@n\MT@ifempty{\the\MT@prot@toks}\relax{\the\MT@prot@toks\MT@gobble@to@nil}}
      {\MT@exp@one@n\MT@prot@l{\the\MT@toks}}}%
  \MT@prot@ifcat{a}{%
    \def\MT@temp*{\MT@prot@addtoken@first}%
  }{%
    \MT@prot@ifcat{!}{%
      \def\MT@temp*{\MT@prot@addtoken@first}%
    }{%
```

a space character, . . .

```
      \MT@prot@ifx\@sptoken{%
        \def\MT@temp* {\MT@prot@get@firstgroup}%
      }{%
```

commands, . . .

```
        \let\MT@prot@ifmacro\MT@prot@ifmacro@
        \MT@map@tlist@c\MT@prot@check@cmds\MT@prot@check
```

. . . or a command/active char whose first command is one of the following:

```
        \MT@prot@ifmacro{%
          \MT@prot@iffirstcmd\UTFviii@two@octets{%
            \def\MT@temp*##1##2{\MT@exp@one@n\MT@prot@l{\the\MT@toks##1##2}}%
          }{%
            \MT@prot@iffirstcmd\UTFviii@three@octets{%
              \def\MT@temp*##1##2##3{\MT@exp@one@n\MT@prot@l{\the\MT@toks##1##2##3}}%
            }{%
              \MT@prot@iffirstcmd\UTFviii@four@octets{%
                \def\MT@temp*##1##2##3##4{\MT@exp@one@n\MT@prot@l{\the\MT@toks##1##2##3##4}}%
              }{%
```

(this is for chars made active by csquotes, via \MakeAutoQuote or \MakeOuterQuote)

```
                \MT@prot@iffirstcmd\csqQQ{\def\MT@temp*##1{\MT@exp@one@n\MT@prot@l{\the\MT@toks##1}}}{%
```

or, finally, a LICR command.

```
                  \MT@prot@iflicrcmd
                }%
              }%
            }%
          }%
        }%
      }%
```

```
    }%
  }%
  \MT@temp*%
}
```

`\MT@prot@addtoken@first` Begin filling toks.

```
\def\MT@prot@addtoken@first#1{%
  \MT@toks\expandafter{\the\MT@toks#1}%
  \MT@prot@get@nexttoken
}
```

`\MT@prot@get@next@token` Continue if letter or other.

```
\def\MT@prot@get@next@token{%
  \def\MT@temp*{\MT@prot@addtoken@next}%
  \MT@prot@ifcat{a}\relax{%
    \MT@prot@ifcat{!}\relax{%
      \def\MT@temp*{\MT@exp@one@n\MT@prot@l{\the\MT@toks}}%
    }%
  }%
  \MT@temp*%
}
⟨/package⟩
```

`\MT@prot@addtoken@next` Add token to our toks and test whether we've seen enough (ligature completed). For luatex, we have to jump through another hoop (i.e., box), because, contrary to the manual, `\lastnodetype` isn't really compatible.

```
⟨*pdf-|lua-|xe-⟩
\def\MT@prot@addtoken@next#1{%
  \MT@toks\expandafter{\the\MT@toks#1}%
  \setbox\MT@tempbox\hbox{%
    \begingroup
```

We disable italic correction, which would prevent us from seeing the ligature (with text commands).

```
    \let\maybe@ic\relax
    \MT@exp@one@n\MT@maybe@textcmd{\the\MT@toks}%
⟨pdf-|xe-⟩     \relax
    \endgroup
⟨lua-⟩  }\setbox\MT@tempbox\hbox{\unhbox\MT@tempbox
    \ifnum\lastnodetype=7 \aftergroup\@firstoftwo\else\aftergroup\@secondoftwo\fi}%
      \MT@prot@get@nexttoken
      {\MT@exp@one@n\MT@prot@l{\the\MT@toks}}%
}
⟨/pdf-|lua-|xe-⟩
```

`\MT@prot@check`
`\MT@prot@check@` We map through a list of commands that should be copied into the toks. ⟨*#3*⟩ will be `\relax` by default, but can also indicate a replacement command.

```
⟨*package⟩
\def\MT@prot@check#1{\MT@prot@check@#1\relax\@nil}
\def\MT@prot@check@#1#2#3\@nil{%
  \ifx\MT@prot@next#2%
    \csname MT@prot@check@#1\endcsname #3%
    \let\MT@prot@ifmacro\@gobble
    \expandafter\MT@tlist@break
  \fi
}
```

Beware that the following nomenclature is rather arcane.

`\MT@prot@check@I`
- This is for commands to be **I**gnored.

```
\def\MT@prot@check@I{%
  \def\MT@temp*##1{\MT@prot@get@firstgroup}%
}
```

\MT@prot@check@S

- Add a **S**ingle command (without an argument).

```
\def\MT@prot@check@S{%
  \def\MT@temp*##1{\MT@toks\expandafter{\the\MT@toks##1}\MT@prot@get@firstgroup}%
}
```

\MT@prot@check@O

- Add a command with **O**ne argument.

```
\def\MT@prot@check@O{%
  \def\MT@temp*##1##2{\MT@toks\expandafter{\the\MT@toks##1{##2}}\MT@prot@get@firstgroup}%
}
```

\MT@prot@check@o
\MT@prot@check@o@

- The same with an optional argument.

```
\def\MT@prot@check@o{%
  \def\MT@temp*##1{\@ifnextchar[{\MT@prot@check@o@##1}{\MT@prot@check@o@##1[]}}%
}
```

The `\color` command, for which this is used, would stumble over an empty optional argument.

```
\def\MT@prot@check@o@#1[#2]#3{%
  \MT@ifempty{#2}
    {\MT@toks\expandafter{\the\MT@toks#1{#3}}}
    {\MT@toks\expandafter{\the\MT@toks#1[#2]{#3}}}%
  \MT@prot@get@firstgroup
}
```

\MT@prot@check@T

- Add a command with **T**wo arguments.

```
\def\MT@prot@check@T{%
  \def\MT@temp*##1##2##3{\MT@toks\expandafter{\the\MT@toks##1{##2}{##3}}\MT@prot@get@firstgroup}%
}
```

\MT@prot@check@E

- This is for commands that **E**nclose their argument in something, e.g., in braces, and which we trial-typeset without any contents.

```
\def\MT@prot@check@E{%
  \the\MT@toks
  \def\MT@temp*##1{\MT@prot@l{##1}}%
}
```

\MT@prot@check@e

- Same for starred commands (the main candidate here is `csquotes`'s `\enquote`).

```
\def\MT@prot@check@e{%
  \the\MT@toks
  \def\MT@temp*##1{\@ifstar{\MT@prot@l{##1*}}{\MT@prot@l{##1}}}%
}
```

\MT@prot@check@eX

- Here we replace the 'integrated interface' (`csquotes`) with the regular one.

```
\def\MT@prot@check@eX#1{%
  \the\MT@toks
  \def\MT@temp*##1{\@ifstar
    {\MT@get@prot{#1*}{left}##1*}
    {\MT@get@prot{#1}{left}##1}}%
}
```

\MT@prot@check@l
\MT@prot@check@l@

- `csquotes` provides a couple of commands for quotations in foreign **l**anguages (lowercase, because it may be starred), whose first argument (the language) we also have to evaluate before trial typesetting.

```
\def\MT@prot@check@l{%
  \def\MT@temp*##1{\@ifstar{\MT@prot@check@l@{##1*}}{\MT@prot@check@l@{##1}}}%
}
\def\MT@prot@check@l@#1#2{%
  \the\MT@toks
  \MT@prot@l{#1{#2}}%
}
```

\MT@prot@check@lX
\MT@prot@check@lX@

- Another macro for `csquotes` commands: replace integrated language-switching commands with their regular variants.

```
\def\MT@prot@check@lX#1{%
  \def\MT@temp*##1{\@ifstar
    {\def\MT@temp{##1*}\MT@prot@check@lX@{#1*}}
    {\def\MT@temp{##1}\MT@prot@check@lX@{#1}}}%
}
\def\MT@prot@check@lX@#1#2{%
  \the\MT@toks
  \MT@get@prot{#1{#2}}{left}\MT@temp{#2}%
}
```

\MT@prot@check@F
\MT@prot@check@F@@

- Here we deal with **F**ont switching commands (i.e., text commands, which take an argument). We (a) remember the text command, (b) save the full text, and then (c) continue inspecting the contents of the argument. We also have to execute (and empty) `\MT@toks`, because it might already contain other commands. Nested text commands still don't work.

```
\def\MT@prot@check@F{%
  \ifx\MT@prot@l\MT@prot@l@tc
    \def\MT@temp*{\MT@exp@one@n\MT@prot@l{\the\MT@toks}}%
  \else
    \let\MT@prot@l\MT@prot@l@tc
    \let\MT@prot@get@first@group\MT@prot@get@first@group@tc
    \def\MT@temp*##1{%
      \the\MT@toks
      \MT@toks{}%
      \MT@prot@check@F@##1%
    }%
  \fi
}
\def\MT@prot@check@F@@#1#2{%
  \let\MT@maybe@textcmd#1%
  \MT@prot@toks{#1{#2}}%
  \MT@prot@get@firstgroup@tc#2\@empty\MT@nil
}
```

\MT@prot@check@F@beamer
\MT@prot@check@F@beamer@

Compatibility with the `beamer` class and its overlay specifications (e.g., `\textbf<2>{...}`).

```
\def\MT@prot@check@F@beamer#1{%
  \@ifnextchar<%
    {\MT@prot@check@F@beamer@#1}%
    {\MT@prot@check@F@@#1}%
}
\def\MT@prot@check@F@beamer@#1<#2>#3{%
  \def\MT@maybe@textcmd{#1<#2>}%
  \MT@prot@toks{#1<#2>{#3}}%
  \MT@prot@get@firstgroup@tc#3\@empty\MT@nil
}
```

\MT@prot@check@F@

Choose the right definition.

```
\@ifclassloaded{beamer}
  {\let\MT@prot@check@F@\MT@prot@check@F@beamer}
  {\let\MT@prot@check@F@\MT@prot@check@F@@}
```

\MT@prot@check@C
\MT@prot@check@C@

- Same, but for commands that allow an optional argument (e.g., the **C**ase changing commands since LaTeX 2022/11/01).

```
\def\MT@prot@check@C{%
  \ifx\MT@prot@l\MT@prot@l@tc
    \def\MT@temp*{\MT@exp@one@n\MT@prot@l{\the\MT@toks}}%
  \else
    \let\MT@prot@l\MT@prot@l@tc
    \let\MT@prot@get@first@group\MT@prot@get@first@group@tc
    \def\MT@temp*##1{%
      \the\MT@toks
      \MT@toks{}%
      \@ifnextchar[%
        {\MT@prot@check@C@##1}%
        {\MT@prot@check@C@##1[]}%
    }%
  \fi
}
\def\MT@prot@check@C@#1[#2]#3{%
  \def\MT@maybe@textcmd{#1[#2]}%
  \MT@prot@toks{#1[#2]{#3}}%
  \MT@prot@get@firstgroup@tc#3\@empty\MT@nil
}
```

\MT@prot@check@cmds

And here’s the list of commands that we can deal with. (It’s a bit of a shame that \textls is not among them.)

```
\def\MT@prot@check@cmds{%
  {I\ignorespaces}{I\relax}{I\@empty}%
  {S\rmfamily}{S\sffamily}{S\ttfamily}{S\mdseries}{S\bfseries}%
  {S\upshape}{S\itshape}{S\slshape}{S\scshape}{S\em}%
  {S\normalfont}{S\selectfont}%
  {S\lsstyle}%
  {S\tiny}{S\scriptsize}{S\footnotesize}{S\small}{S\normalsize}%
  {S\large}{S\Large}{S\LARGE}{S\huge}{S\Huge}%
  {O\fontencoding}{O\fontfamily}{O\fontseries}{O\fontshape}%
  {O\microtypesetup}{O\microtypecontext}%
  {T\fontsize}%
  {F\textrm}{F\textsf}{F\texttt}{F\textnormal}%
  {F\textbf}{F\textmd}{F\textit}{F\textsl}{F\textsc}{F\textup}{F\emph}%
}
```

LaTeX 2020/02/02 introduced some more text commands (adopted from fontaxes, which provides some more, see below).

```
\IfFormatAtLeastTF{2020/02/02}
  {\g@addto@macro\MT@prot@check@cmds{%
    {S\swshape}{S\ulcshape}{S\sscshape}{S\normalshape}%
    {F\textulc}{F\textsw}{F\textssc}%
    {O\fontseriesforce}{O\fontshapeforce}}}
  \relax
\IfFormatAtLeastTF{2022/11/01}
  {\g@addto@macro\MT@prot@check@cmds{{C\MakeUppercase}{C\MakeLowercase}{C\MakeTitlecase}}}
  {\g@addto@macro\MT@prot@check@cmds{{F\MakeUppercase}{F\MakeLowercase}}}
```

The ltxdoc class and the doc package provide some abbreviations. Unfortunately, the \cmd command doesn’t work.

```
\@ifclassloaded{ltxdoc}
  {\g@addto@macro\MT@prot@check@cmds{{E\enquote}{E\marg}{E\oarg}{E\parg}{E\cs}}}\relax
```

Add \color (but not yet \textcolor). We also don’t yet understand beamer’s <.-.> notation added to \color.

```
\MT@addto@setup{%
  \MT@with@package@T{color}
    {\@ifclassloaded{beamer}\relax{\g@addto@macro\MT@prot@check@cmds{{o\color}}}}%
```

csquotes’s \enquote command. It would take precedence over the one provided by ltxdoc.

```
\MT@with@package@T{csquotes}
  {\@ifclassloaded{ltxdoc}
    {\patchcmd\MT@prot@check@cmds{E\enquote}{e\enquote}\relax\relax}
    {\g@addto@macro\MT@prot@check@cmds{{e\enquote}}}%
   \g@addto@macro\MT@prot@check@cmds{{e\textquote}%
     {l\foreignquote}{l\hyphenquote}{l\foreigntextquote}{l\hyphentextquote}%
     {{eX}\textcquote\textquote}%
     {{lX}\foreigntextcquote\foreigntextquote}%
     {{lX}\hyphentextcquote\hyphentextquote}}}%
\MT@with@package@T{doc}
  {\g@addto@macro\MT@prot@check@cmds{{E\meta}}}%
```

The additional fontaxes commands.

```
\MT@with@package@T{fontaxes}
  {\g@addto@macro\MT@prot@check@cmds{%
    {S\txfigures}{S\lnfigures}{S\tbfigures}{S\prfigures}%
    {O\fontfigurestyle}{O\fontfigurealignment}{O\fontbasefamily}%
    {O\figureversion}%
    {F\textfigures}{F\liningfigures}{F\tabularfigures}{F\proportionalfigures}}%
    \IfFormatAtLeastTF{2020/02/02}\relax
      {\g@addto@macro\MT@prot@check@cmds{%
        {S\swshape}{S\ulcshape}{S\sscshape}%
        {F\textulc}{F\textsw}{F\textssc}}}}%
```

fontspec's \fontspec's command allows an optional argument *after* the mandatory one, and we can't deal with that (yet).

```
\MT@with@package@T{fontspec}
  {\g@addto@macro\MT@prot@check@cmds{%
    {O\addfontfeature}{O\addfontfeatures}{F\strong}}}%
```

The nfssext-cfr package (an extension of the nfssext package, which is part of Philipp Lehman's fontinstallationguide but was never publicised separately as far as I can tell) adds many more commands on top of the NFSS.

```
\MT@with@package@T{nfssext-cfr}
  {\g@addto@macro\MT@prot@check@cmds{%
    {S\tistyle}{S\ltstyle}{S\ofstyle}{S\altstyle}{S\regstyle}{S\embossstyle}%
    {S\ornamentalstyle}{S\qtstyle}{S\shstyle}{S\tmstyle}{S\tvstyle}{S\swashstyle}%
    {S\lnstyle}{S\osstyle}{S\instyle}{S\sustyle}{S\lstyle}{S\ostyle}%
    {S\pstyle}{S\tstyle}{S\plstyle}{S\postyle}{S\tlstyle}{S\tostyle}%
    {S\scolshape}{S\olshape}{S\sishape}{S\ushape}{S\scushape}%
    {S\uishape}{S\rishape}{S\dfshape}{S\swstyle}%
    {S\nwwidth}{S\cdwidth}{S\ecwidth}{S\ucwidth}%
    {S\etwidth}{S\epwidth}{S\exwidth}{S\uxwidth}{S\regwidth}%
    {S\mbweight}{S\dbweight}{S\sbweight}{S\ebweight}%
    {S\ubweight}{S\lgweight}{S\elweight}{S\ulweight}%
    {F\textti}{F\textlt}{F\textof}{F\textalt}{F\textreg}{F\emboss}%
    {F\textorn}{O\ornament}{F\textqt}{F\textsh}{F\texttm}{F\texttv}{F\textswash}%
    {F\textln}{F\textos}{F\textin}{F\textsu}{F\textl}{F\texto}%
    {F\textp}{F\textt}{F\textpl}{F\textpo}{F\texttl}{F\textto}%
    {F\textol}{F\textsi}{F\textu}{F\textscu}%
    {F\textui}{F\textri}{F\textdf}%
    {F\textnw}{F\textcd}{F\textec}{F\textuc}%
    {F\textet}{F\textep}{F\textex}{F\textux}{F\textrw}%
    {F\textmb}{F\textdb}{F\textsb}{F\texteb}%
    {F\textub}{F\textlg}{F\textel}{F\textul}}%
    \IfFormatAtLeastTF{2020/02/02}\relax
      {\g@addto@macro\MT@prot@check@cmds{{S\swshape}{F\textsw}}}}%
```

If yfonts is loaded, we add the relevant commands.

```
\MT@with@package@T{yfonts}
  {\g@addto@macro\MT@prot@check@cmds{%
    {S\frakfamily}{S\swabfamily}{S\gothfamily}%
    {F\textfrak}{F\textswab}{F\textgoth}}}%
```

```
}
⟨/package⟩
```

1.2.3 Expansion

\MT@expansion Set up for expansion?

```
⟨*pdf-|lua-⟩
\def\MT@expansion{\MT@maybe@do{ex}}
```

\MT@set@ex@codes@s Setting up font expansion is a bit different because of the selected option. There are two versions of this macro.

If selected=true, we only apply font expansion to those fonts for which a list has been declared (i.e., like for protrusion).

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

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

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

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

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

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

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

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

```
⟨*lua-⟩
\MT@requires@luatex3{
```

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

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

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

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

\MT@reset@ef@codes

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

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

\MT@ex@split@val

There's only one number per character.

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

Take an optional factor into account.

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

Heirs, heirs, I love thy heirs.

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

\MT@warn@ex@too@large

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

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

Apply different values to this font?

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

\MT@get@ex@opt@

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

\MT@set@ex@heirs

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

\MT@preset@ex

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

1.2.4 Interword spacing (glue)

\MT@spacing

Adjustment of interword spacing? Only works with pdfTEX.

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

\MT@set@sp@codes

This is all the same.

```
\def\MT@set@sp@codes{%
```

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

\MT@sp@split@val

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

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

\MT@set@sp@heirs

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

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

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

\MT@preset@sp

\MT@preset@sp@

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

1.2.5 Additional kerning

\MT@kerning

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

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

\MT@set@kn@codes

It’s getting boring, I know.

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

\MT@kn@split@val

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

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

\MT@set@kn@heirs

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

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

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

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

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

1.2.6 Tracking

This only works with pdfTeX 1.40 or LuaTeX 0.62.

```
⟨*pdf-|lua-|xe-⟩
⟨pdf-⟩\MT@requires@pdftex6
⟨lua-⟩\MT@requires@luatex3
⟨pdf-|lua-⟩{
```

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

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

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

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

Tracking won't work with older pdfTEX versions (< 1.40.23) if the original font's \fontdimen 6 is zero, in which case we issue a warning (once for every font).

```
⟨*pdf-|lua-|xe-|letterspace⟩
\def\MT@set@tr@codes{%
⟨*pdf-|lua-|xe-⟩
  \MT@vinfo{Tracking font `\MT@@font'\on@line}%
⟨*pdf-⟩
  \MT@requires@pdftex8\@firstofone{%
    \MT@ifdefined@n@TF{\MT@@font-fake6}{%
      \MT@exp@cs\ifx{\MT@@font-fake6}\@empty
        \MT@warning@nl{%
          Font `\MT@@font' does not specify its\MessageBreak
          \@backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
          tracking will not work with this font}%
        \MT@glet@nc{\MT@@font-fake6}\relax
      \fi
    }%
  }{%
⟨/pdf-⟩
  \MT@if@list@exists
    \MT@get@tr@opt
    \relax
⟨/pdf-|lua-|xe-⟩
  \MT@ifdefined@c@TF\MT@letterspace@\relax{\let\MT@letterspace@\MT@letterspace}%
  \ifnum\MT@letterspace@=\z@
```

Zero tracking requires special treatment.

```
    \MT@set@tr@zero
  \else
⟨pdf-|lua-|xe-⟩   \MT@vinfo{... Tracking by \number\MT@letterspace@}%
```

Letterspacing only works in PDF mode.

```
⟨pdf-|lua-|letterspace⟩   \MT@warn@tracking@DVI
```

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

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

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

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

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

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

```
⟨*lua-|xe-|letterspace⟩
      \MT@if@opentype@font{%
⟨debug⟩\MT@dinfo@nl{1}{... opentype font: \MessageBreak
⟨debug⟩       \expandafter\fontname\font@name}%
⟨lua-|xe-|letterspace⟩       \let\MT@tr@features\@empty
⟨lua-|xe-⟩       \MT@ifdefined@c@T\MT@tr@feat\MT@tr@set@features
        \global\expandafter\font\MT@lsfont=\MT@ls@fontspec@font
⟨debug⟩\MT@dinfo@nl{2}{... -- new font: \expandafter\fontname\MT@lsfont}%
      }{%
⟨/lua-|xe-|letterspace⟩
⟨lua-&debug⟩\MT@dinfo@nl{1}{... legacy font}%
⟨lua-⟩        \MT@ifdefined@c@T\MT@tr@feat
⟨lua-⟩          {\MT@warning{\MT@@font\space is a legacy font.\MessageBreak
⟨lua-⟩                Cannot disable Opentype `features' in \MT@curr@list@name}}%
⟨pdf-|lua-|letterspace⟩      \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@
⟨xe-⟩        \MT@warning{\MT@@font\space is a legacy font.\MessageBreak
⟨xe-⟩                     Cannot letterspace it}%
⟨xe-⟩        \MT@glet\MT@lsfont\font@name
⟨lua-|xe-|letterspace⟩      }%
```

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

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

Adjust outer kerning (`microtype` only).

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

Disable ligatures (not configurable in `letterspace`, not possible with XƎTEX).

```
      \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures
⟨/pdf-|lua-|xe-⟩
⟨*letterspace⟩
      % no ligatures = {f}
      \MT@if@opentype@font
        {\MT@lua{microtype.noligatures([[\MT@lsfont]],[[\number\numexpr`f]])}}
        {\tagcode\MT@lsfont`f=\m@ne}%
⟨/letterspace⟩
```

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

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

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

```
    \aftergroup\MT@set@lsfont
⟨pdf-|lua-|xe-⟩     \let\MT@font\MT@lsfont
⟨lua-|xe-⟩     \MT@if@opentype@font\MT@font\relax
```

`\MT@set@curr@ls` We need to remember the current letterspacing amount (for `\lslig`).

`\MT@curr@ls`

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

Adjust surrounding spacing and kerning.

`\MT@set@curr@os` We get the current outer spacing and adjust it, then, after the end of the current outer group, set the current outer spacing, again, and adjust.

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

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

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

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

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

\MT@set@curr@ok

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

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

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

```
    \MT@afteraftergroup{%
      \MT@set@curr@os
      \MT@set@curr@ok
      \noexpand\MT@tr@outer@r
    }%
⟨/pdf-|lua-|xe-⟩
  \fi
⟨pdf-⟩   }%
}
```

\MT@afteraftergroup

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

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

\MT@ls@fontspec@font

Add the kernfactor feature to a font loaded by fontspec.

```
⟨*lua-|letterspace⟩
\def\MT@ls@fontspec@font{%
  \MT@lua{microtype.add_ls([[\MT@letterspace@]],[[\MT@tr@features]])}%
}
⟨/lua-|letterspace⟩
⟨*xe-⟩
\def\MT@ls@fontspec@font{\MT@exp@two@c\MT@ls@fontspec@font@\fontname\font@name\MT@nil}
\def\MT@ls@fontspec@font@"#1"#2\MT@nil{\MT@ls@fontspec@font@@#1::\MT@nil#2}
\def\MT@ls@fontspec@font@@#1:#2:#3\MT@nil{%
  "#1:#2letterspace=\strip@pt\dimexpr\MT@letterspace@ pt/10\relax
  ;\MT@tr@features"%
}
⟨/xe-⟩
⟨*luafile⟩
local function add_ls(k,feat)
  local f = tex.fontname(font.current())
  local spec,size = match(f,'^(.+)( at .+)$')
  if not spec then spec = f end
  local q = match(spec,'^"') or ""
  local a,b,c = match(spec,'^'..q..'([^:]+):?([^:]*):?(.*)'..q..'$')
  local ls = "kernfactor=" .. k/1000 .. ';'
  microtype.sprint(q..a..':')
  if (a == "name" or a == "file") then
    microtype.sprint(b..':'..ls..c..feat..q)
  else
    microtype.sprint(ls..b..feat..q)
  end
  if size then
    microtype.sprint(size)
  end
end
microtype.add_ls = add_ls

⟨/luafile⟩
```

\MT@get@tr@opt

Various settings (only for the microtype version).

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

\MT@tr@unit@

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

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

\MT@tr@ispace

\MT@tr@ospace

Adjust interword spacing.

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

\MT@tr@okern

Adjust outer kerning.

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

\MT@tr@ligatures

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

```
  \MT@get@tr@opt@{noligatures} {ligatures}%
⟨lua-|xe-⟩  \MT@get@tr@opt@{features}    {feat}%
}
```

\MT@get@tr@opt@

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

\MT@tr@set@features

With LuaTEX or XƎTEX, Ligatures features may be switched on or off.

```
⟨*lua-|xe-⟩
\def\MT@tr@set@features{%
  \MT@map@clist@c\MT@tr@feat{%
    \MT@ifempty{##1}\relax{%
      \MT@if@false
      \lowercase{\edef\@tempa{##1}}%
      \MT@map@tlist@n{{required}      {rlig}}
                      {{common}       {liga}}
                      {{contextual}   {clig}}
                      {{rare}         {dlig}}
                      {{discretionary}{dlig}}
                      {{historic}     {hlig}}
⟨lua-⟩                      {{tex}          {tlig}}
                     }\MT@tr@set@feature@
      \ifMT@if@\else
⟨*xe-⟩
        \MT@ifstreq{\@tempa}{tex}{%
          \MT@xadd\MT@tr@features{mapping=tex-text;}%
        }{%
          \MT@ifstreq{\@tempa}{texoff}{%
            \MT@xadd\MT@tr@features{mapping=;}%
          }{%
            \MT@ifstreq{\@tempa}{notex}{%
              \MT@xadd\MT@tr@features{mapping=;}%
            }{%
⟨/xe-⟩
              \MT@ifstreq{\@tempa}{resetall}{%
                \MT@xadd\MT@tr@features{+dlig;-dlig;+rlig;-rlig;+liga;-liga;+clig;-clig;+hlig;-hlig;%
⟨lua-⟩                                         +tlig,-tlig;%
⟨xe-⟩                                         mapping=tex-text;%
                                       }%
              }{%
                \MT@warning@nl{Unknown Ligatures feature `##1' in \MT@curr@list@name.
                               Ignoring it}%
              }%
⟨xe-⟩          }}}%
      \fi
    }%
  }%
}
```

\MT@tr@set@feature@
\MT@tr@set@feature@@

```
\def\MT@tr@set@feature@#1{%
  \MT@tr@set@feature@@#1%
}
\def\MT@tr@set@feature@@#1#2{%
  \MT@ifstreq\@tempa{#1}{%
    \MT@tr@set@feature@@@{+#2}%
  }{%
    \MT@ifstreq\@tempa{#1off}{%
      \MT@tr@set@feature@@@{-#2}%
    }{%
      \MT@ifstreq\@tempa{no#1}{%
        \MT@tr@set@feature@@@{-#2}%
      }{%
        \MT@ifstreq\@tempa{#1reset}{%
          \MT@tr@set@feature@@@{+#2;-#2}%
        }\relax
      }%
    }%
  }%
}
\def\MT@tr@set@feature@@@#1{%
  \MT@xadd\MT@tr@features{#1;}%
  \MT@if@true
  \MT@tlist@break
}
⟨/lua-|xe-⟩
```

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

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

\lsstyle Disable the tests whether the font should be letterspaced, then trigger the setup. Only \textls can be used in math mode (\lsstyle may be used inside another text switch, of course). Still, we have to ensure that math fonts are set up again. Setting \glb@currsize globally to \@empty (our previous solution) could throw us into an infinite loop (e.g., with the psnfss packages, via \every@math@size), so we issue \glb@settings instead. However, in certain situations, we may still miss some math fonts, so let's try to also enforce it by emptying \glb@currsize, fingers crossed. The overhead seems small.

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

Now the definitions for the letterspace package with plain TEX.

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

\lslig For Fraktur fonts, some ligatures shouldn't be broken up. This command will temporarily select the base font (making sure to really select the current font) and insert the correct kerning.

```
\DeclareRobustCommand\lslig[1]{%
  {\MT@ifdefined@c@TF\MT@curr@ls{%
     \escapechar\m@ne
⟨plain⟩     \MT@requires@latex2{%
     \xdef\font@name{\csname\curr@fontshape/\f@size\endcsname}%
⟨plain⟩     }\relax%
     \MT@get@ls@basefont
     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
     \kern\MT@outer@kern
     \font@name #1%
     \kern\MT@outer@kern
  }{#1}}%
}
```

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

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

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

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

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

If tracking is switched off in the middle of the document, or if \textls is called with a zero letterspacing amount, we have to retrieve the base font and select it.

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

\MT@tr@noligatures

Since an empty value is somewhat ambiguous, we also allow the values 'all' and 'none'. pdfTEX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

```
⟨*pdf-|lua-|xe-⟩
⟨pdf-⟩\MT@requires@pdftex7{
⟨*pdf-|lua-⟩
  \def\MT@tr@noligatures{%
    \ifx\MT@tr@ligatures\@empty
      \MT@noligatures@\MT@lsfont\@undefined
    \else
      \MT@ifstreq\MT@tr@ligatures{all}{%
        \MT@noligatures@\MT@lsfont\@undefined
      }{%
        \MT@ifstreq\MT@tr@ligatures{none}\relax{%
          \MT@noligatures@\MT@lsfont\MT@tr@ligatures
        }%
      }%
    \fi
  }
⟨/pdf-|lua-⟩
⟨*pdf-|xe-⟩
⟨pdf-⟩}{
  \def\MT@tr@noligatures{%
    \MT@ifstreq\MT@tr@ligatures{all}\relax{%
      \MT@warning@nl{%
        Disabling (selected) ligatures is
⟨pdf-⟩        possible since\MessageBreak pdftex 1.40.4.
⟨pdf-⟩        Disabling all ligatures instead%
⟨xe-⟩        not possible with\MessageBreak xetex.
⟨xe-⟩        Ignoring `no ligatures' key in \MT@curr@list@name
      }%
      \MT@glet\MT@tr@noligatures\relax
    }%
  }
⟨pdf-⟩}
⟨/pdf-|xe-⟩
```

\MT@outer@space

A new skip for outer spacing.

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

\MT@tr@set@space

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

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

\MT@tr@set@space@

If settings for outer spacing ⟨*#2*⟩ don't exist, they will be inherited from the inner spacing settings ⟨*#1*⟩.

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

\MT@tr@set@space@@

If the value is followed by an asterisk, the fontdimen will be scaled by the respective amount, otherwise the value denotes the desired dimension in the respective unit.

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

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

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

\MT@tr@outer@l

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

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

Disable left outer kerning.

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

The ragged2e package sets \spaceskip without glue.

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

\MT@tr@outer@next
\MT@tr@outer@r

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

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

\MT@if@outer@next

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

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

\MT@tr@outer@r@

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

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

```
  \ifmmode \else
```

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

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

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

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

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

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

`xspace` requires special treatment.

```
                      \MT@if@outer@next\xspace{%
                        \def\MT@temp*\xspace{\MT@xspace}%
                      }{%
```

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

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

```
}
```

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

Helper macros for the italic correction mess.

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

\MT@xspace
\MT@xspace@

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

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

For older pdfTEX versions and LuaTEX, throw an error.

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

\textls
\MT@ls@adjust@

This command may be used like the other text commands. The starred version removes kerning on the sides. The optional argument changes the letterspacing factor.

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

\MT@textls
\MT@letterspace@

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

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

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

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

```
\def\MT@ls@adjust@empty{\let\MT@ls@adjust\@empty}
```

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

\MT@ls@too@large

Test whether letterspacing amount is too large.

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

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

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

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

\MT@tr@set@okern@

```
\def\MT@tr@set@okern@#1{%
  \MT@test@ast#1*\@nil{%
    \MT@ifdefined@c@TF\MT@tr@unit@
      {\edef\@tempb{#1}\MT@scale@to@em}
      {\@tempcntb=#1\relax}%
    \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
  }{%
    \MT@ifempty\@tempa{\let\@tempa\@m}\relax
    \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
                      * \fontdimen6\MT@lsfont/2000\relax
  }%
⟨lua-⟩  \MT@if@opentype@font\relax{%
⟨pdf-|lua-⟩  \advance\@tempdima -\dimexpr \MT@letterspace@ sp
⟨pdf-|lua-⟩                      * \fontdimen6\MT@lsfont/2000\relax
⟨lua-⟩  }%
  \edef\MT@temp{\MT@temp{\the\@tempdima}}%
}
⟨/pdf-|lua-|xe-⟩
```

\MT@ls@outer@k

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

```
⟨*pdf-|lua-|xe-|letterspace⟩
\def\MT@ls@outer@k{%
  \ifhmode
    \ifdim\lastkern=-3sp \unkern
      \ifdim\lastkern=3sp \kern-3sp
```

```
        \expandafter\expandafter\expandafter\@gobble
      \else \unkern
        \expandafter\expandafter\expandafter\@firstofone
      \fi
    \else
      \expandafter\@firstofone
    \fi
    {\kern\MT@outer@kern\kern3sp\kern-3sp\relax}%
  \fi
}
⟨/pdf-|lua-|xe-|letterspace⟩
```

1.2.7 Disabling ligatures

\MT@noligatures

The possibility to disable ligatures is a new features of pdfTEX 1.30, and also works with LuaTEX.

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

\MT@noligatures@

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

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

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

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

No 'inputenc' key.

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

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

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

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

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

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

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

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

⟨/luafile⟩
```

1.2.8 Loading the configuration

`\MT@load@list`

Recurse through the lists to be loaded.

```
⟨*package|show⟩
```

```
⟨package⟩\def\MT@load@list#1%
⟨show⟩\def\MTS@load@list#1%
  {\edef\@tempa{#1}%
  \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
  \MT@ifstreq\@tempa\@tempb{%
    \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempa' cannot load itself}{}%
  }{%
    \ifx\@tempb\relax
⟨show⟩    :\par\medskip\leavevmode
    \else
      \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
⟨show⟩        \MTS@printtext{, loading \texttt{\@tempb}}%
        \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list `\@tempb'}%
        \begingroup
          \MT@load@list\@tempb
        \endgroup
        \edef\MT@curr@list@name{%
⟨package⟩                  \@nameuse{MT@abbr@\MT@feat} list \noexpand\MessageBreak
                  `\@tempb'}%
        \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
        \expandafter\MT@set@codes\@tempc,\relax,%
⟨show⟩        \vrule width 4cm height .5pt \\
⟨show⟩        \MTS@printtext{End of list \texttt{\MT@curr@list@name}}%
⟨show⟩        \par\medskip\leavevmode
      }{%
        \MT@error{\@nameuse{MT@abbr@\MT@feat} list `\@tempb' undefined.\MessageBreak
                    Cannot load it from list `\@tempa'}{}%
      }%
    \fi
  }%
}
⟨/package|show⟩
```

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

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

```
⟨*package⟩
\let\MT@file@list\@empty
\def\MT@find@file#1{%
```

Check for existence of the file only once.

```
  \MT@in@clist{#1}\MT@file@list
  \ifMT@inlist@ \else
```

Don't forget that because reading the files takes place inside a group, all commands that may be used there have to be defined globally.

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

\MT@cfg@catcodes

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

OK, now all printable characters up to 127 are 'other'. We hope that letters are always letters and numbers other. (`listings` makes them active, see section 1.1.6.)

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

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

\MT@begin@catcodes

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

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

\MT@end@catcodes

End group if outside configuration file (otherwise relax).

```
\let\MT@end@catcodes\endgroup
```

\MT@get@basefamily

The family name might have a suffix e.g., for expert set (`x`), old style numbers (`j`) swash capitals (`w`) etc. We mustn't simply remove the last letter, as this would make for instance `cms` out of `cmss` *and* `cmsy` (OK, `cmex` will still become `cme` . . .).

We only work on the font name if it is longer than three characters.

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

\MT@get@basefamily@

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

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

Table 1:

Order for matching font attributes

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

```
}
```

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

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

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

\MT@try@order

Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don't need table 1 in the documentation part any longer and can cast it off here.

```
\def\MT@try@order{%
  {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
  {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
}
```

\MT@next@listname

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

```
\def\MT@next@listname#1#2#3#4{%
  \ifnum#1=\z@\MT@nofamilytrue\fi
  \edef\@tempa{\MT@encoding
/\ifnum#1=\@ne \MT@family  \fi
/\ifnum#2=\@ne \MT@series  \fi
/\ifnum#3=\@ne \MT@shape   \fi
/\ifnum#4=\@ne *\fi
               \MT@context}%
⟨debug⟩\MT@dinfo@nl{1}{trying \@tempa}%
  \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
    \MT@next@listname@#4%
  }{%
```

Also try with an alias family.

```
    \ifnum#1=\@ne
      \ifx\MT@familyalias\@empty \else
        \edef\@tempa{\MT@encoding
                    /\MT@familyalias
      /\ifnum#2=\@ne \MT@series\fi
      /\ifnum#3=\@ne \MT@shape\fi
      /\ifnum#4=\@ne *\fi
                     \MT@context}%
⟨debug⟩\MT@dinfo@nl{1}{(alias) \@tempa}%
        \MT@ifdefined@n@T{MT@\@tempb @\@tempa}{%
          \MT@next@listname@#4%
        }%
      \fi
    \fi
  }%
}
```

\MT@next@listname@

If size is to be evaluated, do that, otherwise use the current list.

```
\def\MT@next@listname@#1{%
  \ifnum#1=\@ne
    \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
    \ifMT@inlist@
      \let\MT@listname\MT@size@name
    \fi
  \else
    \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
  \fi
}
```

\MT@if@list@exists

\MT@context

```
\def\MT@if@list@exists{%
  \MT@let@cn\MT@context{MT@\MT@feat @context}%
  \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
  \MT@get@listname{\MT@feat @c}%
  \MT@ifdefined@c@TF\MT@listname{%
    \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
    \ifMT@nonselected
      \MT@vinfo{... Applying non-selected expansion (list `\MT@listname')}%
    \else
      \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list `\MT@listname'}%
    \fi
    \@firstoftwo
  }{%
```

Since the name cannot be `\@empty`, this is a sound proof that no matching list exists.

```
    \MT@let@nc{MT@\MT@feat @c@name}\@empty
```

Don’t warn if `selected=false`.

```
    \ifMT@nonselected
      \MT@vinfo{... Applying non-selected expansion (no list)}%
    \else
```

Tracking doesn’t require a list, either.

```
      \MT@ifstreq\MT@feat{tr}\relax{%
        \MT@warning{I cannot find a \@nameuse{MT@abbr@\MT@feat} list
          for font\MessageBreak`\MT@@font'%
            \ifx\MT@context\@empty\else\space(context: `\MT@context')\fi.
          Switching off\MessageBreak\@nameuse{MT@abbr@\MT@feat} for this font}%
      }%
    \fi
    \@secondoftwo
  }%
}
```

\MT@get@inh@list

\MT@context

The inheritance lists are global (no context).

```
\def\MT@get@inh@list{%
  \let\MT@context\@empty

  \MT@get@listname{\MT@feat @inh}%
  \MT@ifdefined@c@TF\MT@listname{%
    \MT@edef@n{MT@\MT@feat @inh@name}{\MT@listname}%
⟨debug⟩\MT@dinfo@nl{1}{... Using \@nameuse{MT@abbr@\MT@feat} inheritance list
⟨debug⟩                `\MT@listname'}%
    \MT@let@cn\@tempc{MT@\MT@feat @inh@\MT@listname}%
```

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

```
    \ifx\@tempc\@empty \else
⟨debug⟩\MT@dinfo@nl{1}{parsing inheritance list ...}%
```

The group is only required in case an input encoding is given.

```
      \begingroup
      \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`\MT@listname'}%
      \MT@set@inputenc{inh}%
      \expandafter\MT@inh@do\@tempc,\relax,%
      \MT@glet@nc{MT@\MT@feat @inh@\MT@listname}\@empty
      \endgroup
    \fi
  }{%
    \MT@let@nc{MT@\MT@feat @inh@name}\@undefined
  }%
}
```

1.2.9 Translating characters into slots

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

\MT@get@slot

There are lots of possibilities how a character may be specified in the configuration files, which makes translating them into slot numbers quite expensive. Also, we want to have this as robust as possible, so that the user does not have to solve a sphinx's riddle if anything goes wrong.

\MT@char

The character is in \@tempa, we want its slot number in \MT@char.

\MT@char@

```
\def\MT@get@slot{%
  \escapechar`\\
  \let\MT@char@\m@ne
  \MT@noresttrue
```

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

```
  \MT@toks=\expandafter{\@tempa}%
```

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

```
  \MT@exp@two@c\MT@is@active\string\@tempa\@nil
```

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

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

```
  \expandafter\MT@is@letter\@tempa\relax\relax
  \ifnum\MT@char@ < \z@
```

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

  If \⟨*encoding*⟩\⟨*command*⟩ (that's *one* command) is defined, we try to extract the slot number.

  We must be cautious not to stumble over accented characters consisting of two commands, like \'\i or \U\CYRI, hence, \string wouldn't be safe enough.

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

- Now, we'll catch the rest, which hopefully is an accented character (e.g. \"a).

```
      {\expandafter\MT@is@composite\@tempa\relax\relax}%
    \ifnum\MT@char@ < \z@
```

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

```
      \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
          \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
    \fi
  \fi

  \let\MT@char\MT@char@
  \MT@get@slot@
  \escapechar\m@ne
}
⟨/package⟩
```

\MT@get@slot@

```
⟨*pdf-|lua-|xe-⟩
\def\MT@get@slot@{%
```

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

```
⟨xe-⟩  \ifnum\XeTeXfonttype\MT@font=\z@
  \ifnum\MT@char > \m@ne
```

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

```
⟨*lua-⟩
    \ifnum\MT@char=47\relax
      \ifMT@norest \else
        \@tempcnta=\MT@lua{
           local glyph = microtype.name_to_slot([[\expandafter\@gobble\@tempa]],true)
           if glyph then tex.write(glyph)
           else tex.write(-1)
           end
        }\relax
        \ifnum\@tempcnta<\z@
          \MT@warn@unknown
          \let\MT@char\m@ne
        \else
          \edef\MT@char{\the\@tempcnta}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}%
        \fi
      \fi
    \else
⟨/lua-⟩
```

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

```
    \ifMT@norest \else
      \MT@warn@rest
⟨pdf-|lua-⟩      \let\MT@char\m@ne
⟨xe-⟩      \let\MT@char\@empty
    \fi
⟨lua-⟩    \fi
  \else
    \MT@warn@unknown
⟨xe-⟩    \let\MT@char\@empty
  \fi
⟨*xe-⟩
  \else
```

There are more possibilities for XƎTEX: It may be a Unicode codepoint (prefixed with '`U`') or a glyph name (prefixed with '`/`').[7] We indicate glyph names

[7] This doesn't seem to be documented anywhere, but it has been announced here: https://tug.org/pipermail/xetex/2010-May/016531.html

to \MT@get@charwd by reversing the sign of \MT@char@.

```
  \ifnum\MT@char=47\relax
   \ifMT@norest \edef\MT@char{U47}%
   \else
    \@tempcnta=\XeTeXglyphindex"\expandafter\@gobble\@tempa"\relax
    \ifnum\@tempcnta=\z@
     \MT@warn@unknown
     \let\MT@char\@empty
    \else
     \edef\MT@char{\@tempa\space}%
     \edef\MT@char@{-\the\@tempcnta}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a glyph name (\the\@tempcnta)}%
    \fi
   \fi
  \else
   \ifnum\MT@char > \m@ne
    \ifMT@norest
```

Or, it's a Unicode number, which we mustn't translate into a glyph number, since the latter is font-specific. But we add the 'U' prefix.

```
     \@tempcnta=\XeTeXcharglyph\MT@char\relax
     \ifnum\@tempcnta=\z@
      \MT@info@missing@char
      \let\MT@char\@empty
     \else
⟨debug⟩\MT@dinfo@nl{3}{> (glyph number: \the\@tempcnta,
⟨debug⟩                   glyph name:   \XeTeXglyphname\MT@font\@tempcnta)}%
      \edef\MT@char{U\MT@char}%
     \fi
    \else
     \MT@warn@rest
     \let\MT@char\@empty
    \fi
   \else
    \MT@warn@unknown
    \let\MT@char\@empty
   \fi
  \fi
 \fi
⟨/xe-⟩
}
⟨/pdf-|lua-|xe-⟩
```

This is the lua function to translate glyph name into slot number. Beginning with v2.2, luaotfload provides this function in its API, which we use if available, but (for now, at least) keep the old code for backward compatibility. With HarfBuzz, the return value is not guaranteed to be inside the Unicode range, so we have to guard against this case as well (same as in do_font). Also, older versions of luaotfload (until v3.18) returned the numbers as floats.

```
⟨*luafile⟩
if luaotfload and luaotfload.aux and luaotfload.aux.slot_of_name then
 local slot_of_name = luaotfload.aux.slot_of_name
 microtype.name_to_slot = function(name, unsafe)
  local n = slot_of_name(font.current(), name, unsafe)
  if not n then return -1 end
  if n > 1114111 then return -1 end
  return math.tointeger(n)
 end
else
 -- we dig into internal structure (should be avoided)
 local function name_to_slot(name, unsafe)
  if fonts then
   local unicodes
```

```
      if fonts.ids then      -- legacy luaotfload
        local tfmdata = fonts.ids[font.current()]
        if not tfmdata then return end
        unicodes = tfmdata.shared.otfdata.luatex.unicodes
      else                   -- new location
        local tfmdata = fonts.hashes.identifiers[font.current()]
        if not tfmdata then return end
        unicodes = tfmdata.resources.unicodes
      end
      local unicode = unicodes[name]
      if unicode then -- does the 'or' branch actually exist?
        return type(unicode) == "number" and unicode or unicode[1]
      end
    end
  end
  microtype.name_to_slot = name_to_slot
end

⟨/luafile⟩
```

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

`\MT@max@char` Warning if resulting character or slot number is too large.

`\MT@max@slot`

```
⟨*pdf-|lua-|xe-⟩
\def\MT@max@char
⟨pdf-⟩  {127 }
⟨lua-|xe-⟩  {1114111 }
\def\MT@max@slot
⟨pdf-⟩  {255 }
⟨lua-|xe-⟩  {1114111 }
⟨/pdf-|lua-|xe-⟩
```

`\ifMT@norest` Test whether all of the string has been used up.

```
⟨*package⟩
\newif\ifMT@norest
```

```
\def\MT@is@letter#1#2\relax{%
  \ifcat a\noexpand#1\relax
    \edef\MT@char@{\number`#1}%
    \ifx\\#2\\%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a letter (\MT@char@)}%
    \else
      \MT@norestfalse
    \fi
  \else
    \ifcat !\noexpand#1\relax
      \edef\MT@char@{\number`#1}%
⟨debug⟩\MT@dinfo@nl{3}{> `\the\MT@toks' is a character (\MT@char@)}%
      \ifx\\#2\\%
        \ifnum\MT@char@ > \MT@max@char \MT@warn@ascii \fi
      \else
        \MT@norestfalse
        \expandafter\MT@is@number#1#2\relax\relax
      \fi
    \fi
  \fi
}
```

`\MT@is@number` Numbers may be specified as a three-digit decimal number (`029`), as a hexadecimal number (prefixed with `"`: `"1D`) or as a octal number (prefixed with `'`: `'35`). They must consist of at least three characters (including the prefix), that is, `"F` is not permitted.

```
\def\MT@is@number#1#2#3\relax{%
  \ifx\relax#3\relax \else
    \ifx\relax#2\relax \else
```

```
      \MT@noresttrue
      \if#1"\relax
        \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
⟨debug⟩\MT@dinfo@nl{3}{> ... a hexadecimal number: \MT@char@}%
      \else
        \if#1'\relax
          \def\MT@char@{\number#1#2#3}%
⟨debug⟩\MT@dinfo@nl{3}{> ... an octal number: \MT@char@}%
        \else
          \MT@ifint{#1#2#3}{%
            \def\MT@char@{\number#1#2#3}%
⟨debug⟩\MT@dinfo@nl{3}{> ... a decimal number: \MT@char@}%
          }\MT@norestfalse
        \fi
      \fi
      \ifnum\MT@char@ > \MT@max@slot
        \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
        \let\MT@char@\m@ne
      \fi
    \fi
  \fi
}
```

\MT@is@active

Expand an active character. (This was completely broken in v1.7, and only worked by chance before.) We \set@display@protect to translate, e.g., Ä into \"A, that is to whatever it is defined in the inputenc encoding file.

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

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

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

Unicode handling has changed again with LaTeX 2019/10/01.

```
      \let\UTF@two@octets@noexpand\@empty
      \let\UTF@three@octets@noexpand\@empty
      \let\UTF@four@octets@noexpand\@empty
```

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

```
      \def\UTFviii@defined##1{\ifx ##1\relax
        \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

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

```
      \MT@ifdefined@c@T\PrerenderUnicode
        {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
      \MT@is@active@hook{#1}%
```

The \expandafter hocus-pocus should please newunicodechar.

```
      \edef\x{\endgroup
        \def\noexpand\@tempa{\expandafter\expandafter\expandafter\@empty\@tempa}%
```

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

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

```
  \fi
}
```

\MT@is@active@hook Test for these packages only once (requires etoolbox).

```
\let\MT@is@active@hook\@gobble
^^Q\@gobble
{\catcode`\#=12
\MT@addto@setup{%
```

If a char has been made active by listings's \lstMakeShortInline, we need to retrieve the original meaning, or else make sure that we're seeing a non-active char.

```
  \MT@with@package@T{listings}{%
    \apptocmd\MT@is@active@hook{%
      \MT@ifdefined@n@T{lst@ShortInlineOldCatcode\string#1}{%
        \catcode`#1=\csname lst@ShortInlineOldCatcode\string#1\endcsname\relax
        \ifnum\catcode`#1=\active
          \begingroup
            \catcode`\~\active \lccode`\~`#1%
            \lowercase{\endgroup
              \MT@let@cn~{lst@ShortInlineOldMeaning\string#1}}%
        \else
          \def\@tempa{#1}%
        \fi
      }%
    }{}{}%
  }%
```

Same for \MakeShortVerb of doc/shortvrb (and implicitly memoir).

```
  \MT@if@false
  \MT@with@package@T{doc}\MT@if@true
  \MT@with@package@T{shortvrb}\MT@if@true
  \ifMT@if@\expandafter\@firstofone\else\expandafter\@gobble\fi{%
    \apptocmd\MT@is@active@hook{%
      \MT@ifdefined@n@T{cc\string#1}{%
        \catcode`#1=\csname cc\string#1\endcsname\relax
        \ifnum\catcode`#1=\active
          \begingroup
            \catcode`\~\active \lccode`\~`#1%
            \lowercase{\endgroup
              \MT@let@cn~{ac\string#1}}%
        \else
          \def\@tempa{#1}%
        \fi
      }%
    }{}{}%
  }%
}}
```

\MT@undefined@char For characters not defined in the current input encoding.

```
\def\MT@undefined@char#1{undefined in input encoding ``#1''}
```

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

```
\def\MT@is@symbol{%
  \expandafter\def\expandafter\MT@char\expandafter
      {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
```

Since recently, some glyphs are defined optionally in LaTeX by checking if the glyph actually exists in the font (e.g., \textasteriskcentered).

```
  \expandafter\expandafter\expandafter
    \MT@is@opt@char\MT@char\iffontchar\char\else\fi\relax
  \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
    \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
  \ifnum\MT@char@ < \z@
```

In TU encoding, some commands (currently, `\textquotesingle`, `\textasciigrave` and `\textquotedbl`) are defined by means of the auxiliary macro `\remove@tlig`, which we take care of here.

```
    \expandafter\expandafter\expandafter\MT@is@tlig\MT@char\relax\relax
    \ifnum\MT@char@ < \z@
```

Finally, if it hasn't been defined by `\DeclareTextSymbol`, it could be a letter (e.g., `\i`, when using `frenchpro`).

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

`\MT@is@opt@char`

This seems adventurous, but we're only redefining the text command within the scope of our setup.

```
\def\MT@is@opt@char#1\iffontchar#2\char#3\else#4\fi\relax{%
  \MT@ifempty{#1}{%
    \iffontchar#2%
      \MT@exp@cs\chardef{\MT@encoding\MT@detokenize@c\@tempa}=#3\relax
    \fi
  }\relax
}
```

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

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

```
\begingroup
  \catcode`\/=\z@
  /MT@map@tlist@n{/\CHARLEX}/@makeother
  /lowercase{%
    /def/x{/endgroup
      /def/MT@charstring{\CHAR"}%
      /def/MT@is@char##1\CHAR"##2##3##4/relax{%
        /ifx/relax##4/relax
          /ifMT@xunicode
            /expandafter/MT@is@charx/MT@strip@prefix##1>/relax\CHAR "%
              /relax/relax/relax/relax/relax
          /fi
        /else
          /ifx/relax##1/relax
            /if##3\/relax
              /edef/MT@char@{/number"##2}%
              /MT@ifstreq/MT@charstring{##3##4}/relax/MT@norestfalse
            /else
              /edef/MT@char@{/number"##2##3}%
              /MT@ifstreq/MT@charstring{##4}/relax
                {/MT@is@xchar##2##3|##4\CHAR"/relax}%
            /fi
⟨debug⟩   /MT@dinfo@nl{3}{> `/the/MT@toks' is a \char (/MT@char@)}%
          /fi
        /fi
      }%
```

`\MT@is@xchar`

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

```
      /def/MT@is@xchar##1|##2\CHAR"##3##4/relax{%
        /MT@ifstreq/MT@charstring{##3##4}%
          {/edef/MT@char@{/number"##1##2}}/MT@norestfalse
      }%
```

`\MT@charxstring`
`\MT@strip@prefix`
`\MT@is@charx`

For `xunicode`, which doesn't `\countdef`, but rather `\def`s the chars.

```
      /def/MT@charxstring{\CHAR "}%
```

```
   /def/MT@strip@prefix##1>##2/relax{##2}%
   /def/MT@is@charx##1\CHAR "##2##3##4##5##6/relax{%
    /ifx/relax##1/relax
     /ifx/relax##6/relax/else
      /edef/MT@char@{/number"##2##3##4##5}%
      /MT@ifstreq{\RELAX >\CHAR "}{##6}/relax/MT@norestfalse
⟨debug⟩    /MT@dinfo@nl{3}{> `/the/MT@toks' is a xunicode \char (/MT@char@)}%
     /fi
    /fi
   }%
  }%
 }
/x
```

\MT@is@tlig

This might have to change again with the next LaTeX release, . . . or so I feared, but it still seems to be fine.

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

\MT@remove@tlig

We remove the `\remove@tlig` command and only pass on the number.

```
\def\MT@remove@tlig{%
 \expandafter\MT@exp@two@c\expandafter\MT@is@number
 \expandafter\@secondoftwo\MT@char\relax\relax
}
```

\MT@is@composite

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

```
\def\MT@is@composite#1#2\relax{%
 \ifx\\#2\\\else
```

Again, we construct a control sequence, this time of the form: `\\`⟨*encoding*⟩ `\`⟨*accent*⟩`-`⟨*character*⟩, e.g., `\\T1\"-a`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\string`ifying it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```
  \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
                \string\csname\MT@encoding\endcsname
                \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
```

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

```
  \ifx\UnicodeEncodingName\@undefined\else
   \expandafter\expandafter\expandafter
    \MT@is@uni@comp\MT@char\iffontchar\else\fi\relax
  \fi
  \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
```

Again, `xunicode`.

```
  \ifnum\MT@char@ < \z@
   \ifMT@xunicode
    \edef\MT@char{\MT@exp@two@c\MT@strip@prefix\meaning\MT@char>\relax}%
    \expandafter\MT@exp@two@c\expandafter\MT@is@charx\expandafter
      \MT@char\MT@charxstring\relax\relax\relax\relax\relax
   \fi
  \fi
 \fi
}
```

\MT@is@uni@comp

Helper for \DeclareUnicodeComposite.

```
\def\MT@is@uni@comp#1\iffontchar#2\else#3\fi\relax{%
  \ifx\\#1\\\edef\MT@char{\iffontchar#2\fi}\fi
}
```

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

```
\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode`#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
    \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}
```

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

Some warning messages, for performance reasons separated here.

\MT@curr@list@name

The type and name of the current list, defined at various places.

\MT@set@listname

```
\def\MT@set@listname{%
  \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
    `\@nameuse{MT@\MT@feat @c@name}'}%
}
```

\MT@warn@ascii

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

```
\def\MT@warn@ascii{%
  \MT@warning@nl{Character `\the\MT@toks' (= \MT@char@)
    is outside of ASCII range.\MessageBreak
    You must load the `inputenc' package before using\MessageBreak
    8-bit characters in \MT@curr@list@name}%
}
```

\MT@warn@number@too@large

Number too large.

```
\def\MT@warn@number@too@large#1{%
  \MT@warning@nl{%
    Number #1 in encoding `\MT@encoding' too large!\MessageBreak
    Ignoring it in \MT@curr@list@name}%
}
```

\MT@warn@rest

Not all of the string has been parsed.

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

\MT@warn@unknown

No idea what went wrong.

```
\def\MT@warn@unknown{%
  \MT@warning@nl{%
    Unknown slot number of character\MessageBreak`\the\MT@toks'%
    \MT@warn@maybe@inputenc\MessageBreak
    in font encoding `\MT@encoding' in \MT@curr@list@name}%
}
```

\MT@warn@maybe@inputenc

In case an input encoding had been requested.

```
\def\MT@warn@maybe@inputenc{%
  \MT@ifdefined@n@T
     {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}%
    { (input encoding `\@nameuse
     {MT@\MT@feat @\MT@cat @\csname MT@\MT@feat @\MT@cat @name\endcsname @inputenc}')}%
}
```

1.2.10 Hook into LaTeX's font selection

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

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

For my reference:

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

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

\MT@font@list

We use a comma separated list.

\MT@font

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

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

```
⟨/package⟩
```

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex2{
\MT@addto@setup{%
```

\MT@orig@pickupfont

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

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

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

```
  \@ifpackageloaded{CJK}{%
```

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

```
    \@ifpackageloaded{xeCJK}{\@firstofone}{%
      \@ifpackagelater{CJK}{2006/10/17}% 4.7.0
        {\def\MT@orig@pickupfont{\CJK@ifundefined\CJK@plane}}%
        {\def\MT@orig@pickupfont{\@ifundefined{CJK@plane}}}%
      \g@addto@macro\MT@orig@pickupfont
        {{\expandafter\ifx\font@name\relax\define@newfont\fi}}%
```

CJKutf8 redefines \pickup@font once more (recent versions, in PDF mode, as determined by ifpdf, which CJKutf8 loads).

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

This is the normal LaTeX definition.

```
  {\def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}}%
```

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

```
  \ifx\pickup@font\MT@orig@pickupfont \else
    \MT@warning@nl{%
```

```
      Command \string\pickup@font\space is not defined as expected.%
      \MessageBreak Patching it anyway. Some things may break%
⟨*package⟩
     .\MessageBreak Double-check whether micro-typography is indeed%
      \MessageBreak applied to the document.%
      \MessageBreak (Hint: Turn on `verbose' mode)%
⟨/package⟩
    }%
  \fi
```

\pickup@font

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

```
  \g@addto@macro\pickup@font{\begingroup}%
```

If the trace package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.

```
  \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}%
  \g@addto@macro\pickup@font{%
```

If we're inside an \edef (or \write ...), we don't want to execute our code. This will still leave '\begingroup \let \relax \relax \endgroup' in the input stream, which is not nothing but should be harmless enough. \pickup@font should never be executed in these contexts anyway, but obviously this may, under rare circumstances, still happen (e.g., with hyperref).[8]

```
⟨package⟩    \MT@if@expanding@F{%
      \escapechar\m@ne
⟨*package⟩
⟨debug⟩      \global\MT@inannottrue
⟨debug⟩      \MT@glet\MT@pdf@annot\@empty
⟨debug⟩      \MT@addto@annot{(line \number\inputlineno)}%
```

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

```
      \MT@let@cn\MT@font{MT@subst@\expandafter\string\font@name}%
      \ifx\MT@font\relax
        \let\MT@font\font@name
      \else
        \ifx\MT@font\font@name \else
⟨debug⟩  \MT@addto@annot{= substituted with \MT@@font}%
          \MT@register@subst@font
        \fi
      \fi
      \MT@setupfont}%
⟨/package⟩
⟨letterspace⟩      \MT@tracking
    \endgroup
  }%
⟨*package⟩
```

\MT@pickupfont
\MT@MT@pickupfont
\MT@ltx@pickupfont

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

```
  \let\MT@pickupfont\pickup@font
  \def\MT@MT@pickupfont {\let\pickup@font\MT@pickupfont}%
  \def\MT@ltx@pickupfont{\let\pickup@font\MT@orig@pickupfont}%
```

\do@subst@correction

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

[8] Cf. https://tex.stackexchange.com/q/687763/7674

the document, not just for the first time it is called, since we need it every time a font is letterspaced.

```
\g@addto@macro\do@subst@correction
  {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
   \MT@glet@nc{MT@subst@\expandafter\string\font@name}\MT@font}%
```

\add@accent
\MT@orig@add@accent

Inside \add@accent, we have to disable microtype's setup, since the grouping in the patched \pickup@font would break the accent if different fonts are used for the base character and the accent. Fortunately, LaTeX takes care that the fonts used for the \accent are already set up, so that we cannot be overlooking them.

```
\let\MT@orig@add@accent\add@accent
\def\add@accent#1#2{%
  \MT@ltx@pickupfont
  \MT@orig@add@accent{#1}{#2}%
  \MT@MT@pickupfont
}%
⟨/package⟩
}
⟨plain⟩}\relax
⟨/package|letterspace⟩
⟨*package⟩
```

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

\MT@check@font

Check whether we've already seen the current font.

```
\def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

\MT@register@font

Register the current font.

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

\MT@register@subst@font

Register the substituted font (only if it isn't registered already). Additionally, we have to remove the substitute font from the list of fonts, so that we set it up again.

```
\def\MT@register@subst@font{%
  \MT@exp@one@n\MT@in@clist\font@name\MT@font@list
  \ifMT@inlist@\else
    \xdef\MT@font@list{\MT@font@list\font@name,}%
    \expandafter\MT@rem@from@clist\MT@font\MT@font@list
  \fi
}
⟨/package⟩
```

1.2.11 Context-sensitive setup

Here are the variants for context-sensitive setup.

\MT@active@features

The activated features are stored in a command. We always allow contexts for tracking, because \textls may be used without activating the feature.

```
⟨*pdf-|lua-|xe-⟩
⟨pdf-⟩\MT@requires@pdftex6
⟨lua-⟩\MT@requires@luatex3
⟨pdf-|lua-⟩  {%
  \def\MT@active@features{,tr}%
⟨pdf-|lua-⟩  }{\let\MT@active@features\@empty}
⟨/pdf-|lua-|xe-⟩
```

\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.

```
⟨*package⟩
```

```
\def\MT@check@font@cx{%
  \MT@if@true
  \MT@map@clist@c\MT@active@features{%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
      \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \ifMT@inlist@
      \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
    \else
      \MT@if@false
    \fi
  }%
  \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
}
```

\MT@register@subst@font@cx

Add the substituted font to each feature list and possibly remove substitute font.

```
\def\MT@register@subst@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\font@name
      \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \ifMT@inlist@ \else
      \MT@exp@cs\MT@xadd
        {MT@##1@\csname MT@##1@context\endcsname font@list}%
        {\font@name,}%
      \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter\MT@font
         \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
    \fi
  }%
}
```

\MT@register@font@cx

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

```
\def\MT@register@font@cx{%
  \MT@map@clist@c\MT@active@features{%
    \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
      \MT@exp@cs\MT@xadd
        {MT@##1@\csname MT@##1@context\endcsname font@list}%
        {\MT@font,}%
      \def\@tempa{##1}%
      \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
    \fi
  }%
}
```

\MT@maybe@rem@from@list

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

```
\def\MT@maybe@rem@from@list#1{%
  \MT@ifstreq{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
    \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
       \MT@font \csname MT@\@tempa @#1font@list\endcsname
  }%
}
```

\microtypecontext
\MT@microtypecontext

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

Inside the preamble, this command shouldn't actually do anything but remember itself for later.

```
\DeclareRobustCommand\microtypecontext{\MT@begin@catcodes\MT@microtypecontext}
\def\MT@microtypecontext#1{\MT@end@catcodes\MT@addto@setup{\microtypecontext{#1}}}
\MT@addto@setup{%
  \DeclareRobustCommand\microtypecontext{%
    \MT@begin@catcodes
    \MT@microtypecontext
  }%
  \def\MT@microtypecontext#1{%
    \MT@end@catcodes
    \MT@setup@contexts
    \let\MT@reset@context\relax
```

We need to ensure that math fonts are set up anew.

```
    \MT@glet\glb@currsize\@empty
    \setkeys{MTC}{#1}%
    \selectfont
    \MT@reset@context
  }%
}
```

\textmicrotypecontext
\MT@textmicrotypecontext
\MT@text@microtypecontext

This is just a wrapper around \microtypecontext.

```
\DeclareRobustCommand\textmicrotypecontext{\MT@begin@catcodes\MT@textmicrotypecontext}
\def\MT@textmicrotypecontext#1{\MT@end@catcodes\MT@text@microtypecontext{#1}}
\def\MT@text@microtypecontext#1#2{{\microtypecontext{#1}#2}}
```

\MT@reset@context
\MT@reset@context@

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

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

\MT@setup@contexts

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

```
\def\MT@setup@contexts{%
  \MT@map@clist@c\MT@active@features
    {\MT@glet@nc{MT@##1@@font@list}\MT@font@list}%
  \MT@glet\MT@check@font\MT@check@font@cx
  \MT@glet\MT@register@font\MT@register@font@cx
  \MT@glet\MT@register@subst@font\MT@register@subst@font@cx
  \MT@glet\MT@setup@contexts\relax
}
```

Define context keys.

```
\MT@map@clist@c\MT@features@long{%
  \define@key{MTC}{#1}[]{%
    \edef\@tempb{\@nameuse{MT@rbba@#1}}%
    \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
    \ifMT@inlist@
```

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

```
      \MT@ifempty{##1}{\def\MT@val{@}}{\def\MT@val{##1}}%
      \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
⟨debug⟩\MT@dinfo{1}{>>> no change of #1 context: `\MT@val'}%
      \else
        \MT@vinfo{>>> Changing #1 context to `\MT@val'\MessageBreak\on@line
⟨debug⟩          \space(previous: `\@nameuse{MT@\@tempb @context}')%
                 }%
        \def\MT@reset@context{\aftergroup\MT@reset@context@}%
```

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

```
        \MT@glet@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%
```

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

```
        \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
          \MT@val \csname MT@\@tempb @doc@contexts\endcsname
        \ifMT@inlist@ \else
          \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{{\MT@val}}%
⟨debug⟩  \MT@dinfo{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
```

```
      \fi
      \MT@edef@n{MT@\@tempb @context}{\MT@val}%
    \fi
  \fi
 }%
}
```

We also allow the `activate` shortcut.

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

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

Initialise the contexts.

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

1.3 Configuration

1.3.1 Font sets

\DeclareMicrotypeSet
\DeclareMicrotypeSet*

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

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

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

\MT@DeclareSet

```
\newcommand\MT@DeclareSet[3][]{%
  \MT@ifempty{#1}{%
    \MT@map@clist@c\MT@features{\begingroup\MT@declare@sets{##1}{#2}{#3}\endgroup}%
  }{%
    \MT@map@clist@n{#1}{\begingroup
      \MT@ifempty{##1}\relax{%
        \MT@is@feature{##1}{set declaration `#2'}{%
          \MT@exp@one@n\MT@declare@sets
            {\csname MT@rbba@##1\endcsname}{#2}{#3}%
        }%
      }%
    \endgroup}%
  }%
  \MT@end@catcodes
}
```

\MT@DeclareSetAndUseIt

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

\MT@curr@set@name

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

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

\MT@declare@sets

Define the current set name and parse the keys.

```
\def\MT@declare@sets#1#2#3{%
  \def\MT@curr@set@name{#2}%
  \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
    \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
    \MT@map@clist@n{font,encoding,family,series,shape,size}{%
      \MT@glet@nc{MT@#1list@##1@\MT@curr@set@name}\@undefined
    }%
  }%
  \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\@empty
⟨debug⟩\MT@dinfo{1}{declaring \@nameuse{MT@abbr@#1} set `\MT@curr@set@name'}%
  \setkeys{MT@#1@set}{#3}%
}
```

\MT@define@set@key@

⟨*#1*⟩ = font axis, ⟨*#2*⟩ = feature.

```
\def\MT@define@set@key@#1#2{%
  \define@key{MT@#2@set}{#1}[]{%
    \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\@empty
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@get@highlevel{#1}%
```

We do not add the expanded value to the list . . .

```
      \MT@exp@two@n\g@addto@macro
        {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
        {\MT@val,}%
    }%
```

. . . but keep in mind that the list has to be expanded at the end of the preamble.

```
    \expandafter\g@addto@macro\expandafter\MT@font@sets
      \csname MT@#2list@#1@\MT@curr@set@name\endcsname
⟨debug⟩\MT@dinfo@nl{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
  }%
}
```

\MT@get@highlevel

Saying, for instance, '`family=rm*`' or '`shape=bf*`' will expand to `\rmdefault` resp. `\bfdefault`.

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

And '`family = *`' will become `\familydefault`.

```
    \MT@ifempty\@tempa{\def\@tempa{#1}}\relax
```

Test whether the command is actually defined.

```
    \MT@ifdefined@n@TF{\@tempa default}%
      {\edef\MT@val{\MT@exp@cs\noexpand{\@tempa default}}}%
      {\MT@warning{`\@backslashchar\@tempa default' is not a defined command.\MessageBreak
                   Ignoring `#1 = {\@tempa*}' in font set\MessageBreak`\MT@curr@set@name'}%
       \let\MT@val\@empty}%
```

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

```
  }%
}
```

\MT@test@ast

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

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

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

Fully expand the font specification and fix catcodes for all font sets. Also remove fontspec's counters.

```
\let\MT@font@sets\@empty
\def\MT@fix@font@set#1{%
  \MT@ifdefined@c@T{#1}{%
    \xdef#1{#1}%
    \ifMT@fontspec
      \xdef#1{\expandafter\MT@scrubfeatures#1()\relax}%
    \fi
    \global\@onelevel@sanitize#1%
  }%
}
```

\MT@define@set@key@size

size requires special treatment.

```
\def\MT@define@set@key@size#1{%
  \define@key{MT@#1@set}{size}[]{%
    \MT@map@clist@n{##1}{%
      \def\MT@val{####1}%
      \expandafter\MT@get@range\MT@val--\@nil
      \ifx\MT@val\relax \else
        \MT@exp@cs\MT@xadd
          {MT@#1list@size@\MT@curr@set@name}%
          {{{\MT@lower}{\MT@upper}\relax}}%
      \fi
    }%
⟨debug⟩\MT@dinfo@nl{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%
  }%
}
```

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

\MT@get@range
\MT@upper
\MT@lower

Ranges will be stored as triplets of {⟨*lower bound*⟩}{⟨*upper bound*⟩}{⟨*list name*⟩}. For simple sizes, the upper boundary is -1.

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

2048 pt is TeX's maximum font size.

```
          {\def\MT@upper{2048}}%
      }{%
        \def\MT@val{#2}%
        \MT@get@size
        \ifx\MT@val\relax \else
          \MT@ifdim\MT@lower>\MT@val{%
            \MT@error{%
              Invalid size range (\MT@lower\space > \MT@val) in font set
              `\MT@curr@set@name'.\MessageBreak Swapping sizes}{}%
            \edef\MT@upper{\MT@lower}%
            \edef\MT@lower{\MT@val}%
          }{%
            \edef\MT@upper{\MT@val}%
          }%
          \MT@ifdim\MT@lower=\MT@upper
            {\def\MT@upper{-1}}%
            \relax
        \fi
      }%
    \fi
  }%
}
```

\MT@get@size

Translate a size selection command and normalise it.

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

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

```
  \if*\MT@val\relax
    \def\@tempa{\normalsize}%
  \else
    \MT@let@cn\@tempa{\MT@val}%
  \fi
  \ifx\@tempa\relax\else
    \MT@get@size@
  \fi
```

Font specifications also accept dimens. (\ifdefdimen is provided by etoolbox.)

```
^^X  \MT@exp@one@n\ifdefdimen\MT@val{\edef\MT@val{\the\MT@val}}\relax
```

Test whether we finally got a number or dimension so that we can strip the 'pt' (\@defaultunits and \strip@pt are kernel macros).

```
  \MT@ifdimen\MT@val{%
    \@defaultunits\@tempdima\MT@val pt\relax\@nnil
    \edef\MT@val{\strip@pt\@tempdima}%
  }{%
    \MT@warning{Could not parse font size `\MT@val'\MessageBreak
                in font set `\MT@curr@set@name'}%
    \let\MT@val\relax
  }%
}
```

\MT@get@size@
\MT@get@size@@

The relsize solution of parsing \@setfontsize does not work with the AMS classes, among others. I hope my hijacking doesn't do any harm. We redefine \set@fontsize instead of \@setfontsize because some classes might define the size selection commands by simply using \fontsize (e.g., the a0poster class).

```
\def\MT@get@size@@{%
  \begingroup
    \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
    \@tempa\@nil
}
```

The svjour3 class defines the size commands using conditionals; using e-TeX primitives, we close any leftovers here.

```
^^X\@ifclassloaded{svjour3}{%
^^X  \def\MT@get@size@{%
^^X    \@tempcnta=\currentiflevel
^^X    \MT@get@size@@
^^X    \MT@loop
^^X      \ifnum\numexpr\currentiflevel-1>\@tempcnta
^^X      \csname fi\endcsname
^^X    \MT@repeat
^^X  }%
^^X}{%
  \let\MT@get@size@\MT@get@size@@
^^X}
```

\MT@define@set@key@font

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

\MT@get@font Translate any asterisks.

```
\def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
  \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
  \ifx\MT@val\relax\def\MT@val{0}\fi
  \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
  \let\MT@val\@tempb
}
```

\MT@get@font@ Helper macro, also used by \MT@get@font@and@size.

```
\def\MT@get@font@#1#2#3#4#5#6{%
  \let\@tempb\@empty
  \def\MT@temp{#1/#2/#3/#4/#5}%
  \MT@get@axis{encoding}{#1}%
  \MT@get@axis{family}  {#2}%
  \MT@get@axis{series}  {#3}%
  \MT@get@axis{shape}   {#4}%
  \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
  \MT@ifempty{#5}{%
    \MT@warn@axis@empty{size}{\string\normalsize}%
    \def\MT@val{*}%
  }{%
    \def\MT@val{#5}%
  }%
  \MT@get@size
}
```

\MT@get@axis

```
\def\MT@get@axis#1#2{%
  \def\MT@val{#2}%
  \MT@get@highlevel{#1}%
  \MT@ifempty\MT@val{%
    \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
    \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
  }\relax
  \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%
}
```

\MT@warn@axis@empty

```
\def\MT@warn@axis@empty#1#2{%
  \MT@warning{#1 axis is empty in font specification\MessageBreak
    `\MT@temp'. Using `#2' instead}%
}
```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

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

\UseMicrotypeSet

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

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

\MT@UseMicrotypeSet

```
\newcommand*\MT@UseMicrotypeSet[2][]{%
  \MT@ifempty{#1}{%
    \MT@map@clist@c\MT@features{\begingroup\MT@use@set{##1}{#2}\endgroup}%
  }{%
    \MT@map@clist@n{#1}{\begingroup
      \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}%
        }%
      }%
    \endgroup}%
  }%
  \MT@end@catcodes
}
```

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

Only use sets that have been declared.

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

\DeclareMicrotypeSetDefault

This command can be used in the main configuration file to declare the default font set, in case no set is specified in the package options.

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

\MT@DeclareMicrotypeSetDefault

```
\newcommand*\MT@DeclareMicrotypeSetDefault[2][]{%
  \MT@ifempty{#1}{%
    \MT@map@clist@c\MT@features{\begingroup\MT@set@default@set{##1}{#2}\endgroup}%
  }{%
    \MT@map@clist@n{#1}{\begingroup
```

```
      \MT@ifempty{##1}\relax{%
        \MT@is@feature{##1}{declaration of default set `#2'}{%
          \MT@exp@one@n\MT@set@default@set
            {\csname MT@rbba@##1\endcsname}{#2}%
        }%
      }%
    \endgroup}%
  }%
  \MT@end@catcodes
}
```

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

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

1.3.2 Variants and aliases

\DeclareMicrotypeVariants
\MT@variants

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

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

\MT@DeclareVariants

```
\def\MT@DeclareVariants#1{%
  \MT@map@clist@n{#1}{%
    \def\@tempa{##1}%
    \@onelevel@sanitize\@tempa
    \xdef\MT@variants{\MT@variants{\@tempa}}%
  }%
  \MT@end@catcodes
}
```

\DeclareMicrotypeAlias

This can be used to set an alias name for a font, so that the file and the settings for the aliased font will be loaded.

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

\MT@DeclareMicrotypeAlias

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

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

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

1.3.3 Configuration file management

`\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@cfg@prefix-\@tempa.cfg already loaded}%
  \else
    \MT@xadd\MT@file@list{\@tempa,}%
    \MT@begin@catcodes
    \InputIfFileExists{\MT@cfg@prefix-\@tempa.cfg}{%
      \edef\MT@curr@file{\MT@cfg@prefix-\@tempa.cfg}%
      \MT@vinfo{... Loading configuration file \MT@curr@file}%
    }{%
      \MT@warning{Configuration file \MT@cfg@prefix-\@tempa.cfg\MessageBreak
                  does not exist}%
    }%
    \MT@end@catcodes
  \fi
}
```

`\MT@cfg@prefix`
`\DeclareMicrotypeFilePrefix`

The configuration files’ prefix may be customised.

```
\def\MT@cfg@prefix{mt}
\def\DeclareMicrotypeFilePrefix#1{%
  \def\MT@cfg@prefix{#1}%
}
⟨/package⟩
```

1.3.4 Disabling ligatures

`\DisableLigatures`
`\MT@DisableLigatures`
`\MT@nl@setname`
`\MT@nl@ligatures`

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

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

```
⟨*pdf-|lua-⟩
⟨pdf-⟩\MT@requires@pdftex5{
\def\DisableLigatures{%
  \MT@begin@catcodes
  \MT@DisableLigatures
}
\newcommand*\MT@DisableLigatures[2][]{%
  \MT@ifempty{#1}\relax{\gdef\MT@nl@ligatures{#1}}%
  \xdef\MT@active@features{\MT@active@features,nl}%
  \global\MT@noligaturestrue
  \MT@declare@sets{nl}{no ligatures}{#2}%
  \gdef\MT@nl@setname{no ligatures}%
  \MT@end@catcodes
}
⟨pdf-⟩}{
⟨/pdf-|lua-⟩
```

If pdfTEX is too old, we throw an error.

```
⟨*pdf-|xe-⟩
\renewcommand*\DisableLigatures[2][]{%
  \MT@error{Disabling ligatures of a font is only possible\MessageBreak
    with pdftex version 1.30 or newer.\MessageBreak
    Ignoring \@backslashchar DisableLigatures}{%
⟨pdf-⟩    Upgrade
⟨xe-⟩    Use
    pdftex.}%
}
⟨pdf-⟩}
⟨/pdf-|xe-⟩
```

1.3.5 Interaction with babel

\DeclareMicrotypeBabelHook
\MT@DeclareMicrotypeBabelHook

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

```
⟨*package⟩
\def\DeclareMicrotypeBabelHook{%
  \MT@begin@catcodes
  \MT@DeclareMicrotypeBabelHook
}
\def\MT@DeclareMicrotypeBabelHook#1#2{%
  \MT@map@clist@n{#1}{%
    \KV@@sp@def\@tempa{##1}%
    \MT@gdef@n{MT@babel@\@tempa}{#2}%
  }%
  \MT@end@catcodes
}
```

1.3.6 Fine tuning

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

\SetProtrusion

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

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

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

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

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

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

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

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

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

```
  \MT@permute
```

. . . which we can now define to be ⟨*#3*⟩. Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

```
  \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
  \MT@end@catcodes
}
⟨/package⟩
```

\SetExpansion

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

```
⟨*pdf-|lua-⟩
\def\SetExpansion{%
  \MT@begin@catcodes
  \MT@SetExpansion
}
```

\MT@SetExpansion
\MT@ex@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExpansion[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@ex@c}{#1}%
  \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @factor}{%
    \ifnum\csname MT@ex@c@\MT@ex@c@name @factor\endcsname > \@m
      \MT@warning@nl{Expansion factor \number\@nameuse{MT@ex@c@\MT@ex@c@name @factor}
        too large in list\MessageBreak `\MT@ex@c@name'. Setting it to the
        maximum of 1000}%
      \MT@glet@nc{MT@ex@c@\MT@ex@c@name @factor}\@m
    \fi
  }%
⟨debug⟩\MT@dinfo{1}{creating expansion list `\MT@ex@c@name'}%
  \def\MT@permutelist{ex@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@ex@c@\MT@ex@c@name}{#3}%
  \MT@end@catcodes
}
⟨/pdf-|lua-⟩
```

\SetTracking

```
⟨*pdf-|lua-|xe-⟩
\def\SetTracking{%
  \MT@begin@catcodes
  \MT@SetTracking
}
```

\MT@SetTracking

Third argument may be empty.

```
\newcommand*\MT@SetTracking[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@tr@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating tracking list `\MT@tr@c@name'}%
  \def\MT@permutelist{tr@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \KV@@sp@def\@tempa{#3}%
  \MT@ifempty\@tempa\relax{%
    \MT@ifint\@tempa
      {\MT@xdef@n{MT@tr@c@\MT@tr@c@name}{\@tempa}}%
      {\MT@warning{Value `\@tempa' is not a number in\MessageBreak
                   tracking set `\MT@curr@set@name'}}}%
  \MT@end@catcodes
}
⟨/pdf-|lua-|xe-⟩
```

\SetExtraSpacing

```
⟨*pdf-⟩
\def\SetExtraSpacing{%
```

```
  \MT@begin@catcodes
  \MT@SetExtraSpacing
}
```

\MT@SetExtraSpacing
\MT@sp@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExtraSpacing[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@sp@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating spacing list `\MT@sp@c@name'}%
  \def\MT@permutelist{sp@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@sp@c@\MT@sp@c@name}{#3}%
  \MT@end@catcodes
}
```

\SetExtraKerning

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

\MT@SetExtraKerning
\MT@kn@c@name
\MT@extra@context
\MT@permutelist

```
\newcommand*\MT@SetExtraKerning[3][]{%
  \let\MT@extra@context\@empty
  \MT@set@named@keys{MT@kn@c}{#1}%
⟨debug⟩\MT@dinfo{1}{creating kerning list `\MT@kn@c@name'}%
  \def\MT@permutelist{kn@c}%
  \setkeys{MT@cfg}{#2}%
  \MT@permute
  \MT@gdef@n{MT@kn@c@\MT@kn@c@name}{#3}%
  \MT@end@catcodes
}
⟨/pdf-⟩
```

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

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

```
⟨*package⟩
\def\MT@set@named@keys#1#2{%
  \def\x##1name=##2,##3\@nil{%
    \setkeys{#1}{name=##2}%
    \gdef\MT@options{##1##3}%
    \MT@rem@from@clist{name=}\MT@options
  }%
  \x#2,name=,\@nil
  \@expandtwoargs\setkeys{#1}\MT@options
}
```

\MT@define@code@key

Define the keys for the configuration lists (which are setting the codes, in pdfTEX speak).

```
\def\MT@define@code@key#1#2{%
  \define@key{MT@#2}{#1}[]{%
    \@tempcnta=\@ne
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
```

Here, too, we allow for something like ‘bf*’. It will be expanded immediately.

```
      \MT@get@highlevel{#1}%
      \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
      \advance\@tempcnta \@ne
    }%
  }%
}
```

\MT@define@code@key@family

Remove fontspec's internal feature counter.

```
\def\MT@define@code@key@family#1{%
  \define@key{MT@#1}{family}[]{%
    \@tempcnta=\@ne
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@get@highlevel{family}%
      \ifMT@fontspec
        \edef\x{\edef\noexpand\MT@val{\noexpand\MT@scrubfeature\MT@val()\relax}}\x
      \fi
      \MT@edef@n{MT@tempfamily\the\@tempcnta}{\MT@val}%
      \advance\@tempcnta \@ne
    }%
  }%
}
```

\MT@define@code@key@size

\MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.

```
\def\MT@define@code@key@size#1{%
  \define@key{MT@#1}{size}[]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \expandafter\MT@get@range\MT@val--\@nil
      \ifx\MT@val\relax \else
        \MT@exp@cs\MT@xadd{MT@tempsize}%
           {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
      \fi
    }%
  }%
}
```

\MT@define@code@key@font

```
\def\MT@define@code@key@font#1{%
  \define@key{MT@#1}{font}[]{%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifstreq\MT@val*{\def\MT@val{*/*/*/*/*}}\relax
      \expandafter\MT@get@font@and@size\MT@val/////\@nil
      \ifMT@fontspec
        \edef\@tempb{\expandafter\MT@scrubfeatures\@tempb()\relax}%
      \fi
      \MT@xdef@n{MT@\MT@permutelist @\@tempb\MT@extra@context}%
        {\csname MT@\MT@permutelist @name\endcsname}%
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempb=\MT@val
⟨debug⟩                \ifx\MT@extra@context\@empty\else\MessageBreak
⟨debug⟩                  (context: \MT@extra@context)\fi}%
      \MT@exp@cs\MT@xaddb
        {MT@\MT@permutelist @\@tempb\MT@extra@context @sizes}%
        {{{\MT@val}{\m@ne}{\MT@curr@set@name}}}%
    }%
  }%
}
```

\MT@get@font@and@size

Translate any asterisks and split off the size.

```
\def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
  \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
}
```

```
\MT@define@code@key{encoding}{cfg}
\MT@define@code@key@family   {cfg}
\MT@define@code@key{series}  {cfg}
\MT@define@code@key{shape}   {cfg}
\MT@define@code@key@size     {cfg}
\MT@define@code@key@font     {cfg}
```

\MT@define@opt@key

```
\def\MT@define@opt@key#1#2{%
  \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
    \MT@xdef@n{MT@#1@c@\MT@curr@set@name @#2}{##1}}}%
}
```

\MT@listname@count

The options in the optional first argument.

```
\newcount\MT@listname@count
\MT@map@clist@c\MT@features{%
```

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

```
  \define@key{MT@#1@c}{name}[]{%
    \MT@ifempty{##1}{%
      \MT@ifdefined@n@TF{MT@#1@c@\MT@curr@file/\the\inputlineno}{%
        \global\advance\MT@listname@count\@ne
        \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno
                                 (\number\MT@listname@count)}%
      }{%
        \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
      }%
    }{%
      \MT@edef@n{MT@#1@c@name}{##1}%
      \MT@ifdefined@n@T{MT@#1@c@\csname MT@#1@c@name\endcsname}{%
        \MT@warning{Redefining \@nameuse{MT@abbr@#1} list `\@nameuse{MT@#1@c@name}'}%
      }%
    }%
    \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
  }%
  \MT@define@opt@key{#1}{load}%
  \MT@define@opt@key{#1}{factor}%
  \MT@define@opt@key{#1}{preset}%
  \MT@define@opt@key{#1}{inputenc}%
```

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

```
  \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
}
⟨/package⟩
```

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdfTEX versions, disallow. It also works with LuaTEX 0.30 or newer.

```
⟨*pdf-|lua-⟩
⟨pdf-⟩\MT@requires@pdftex7{
  \define@key{MT@ex@c}{context}[]{%
    \MT@ifempty{#1}\relax{%
      \MT@glet\MT@copy@font\MT@copy@font@
      \def\MT@extra@context{#1}%
    }%
  }
  \MT@addto@setup{%
    \define@key{MT@ex@c}{context}[]{%
      \ifx\MT@copy@font\MT@copy@font@
        \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
      \else
        \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
           Ignoring `context' key\on@line}%
          {Either move the settings inside the preamble,\MessageBreak
           or load the package with the `copyfonts' option.}%
      \fi
    }%
  }
```

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

```
  \define@key{MT@pr@c}{context}[]{%
    \MT@ifempty{#1}\relax{%
      \MT@glet\MT@copy@font\MT@copy@font@
      \def\MT@extra@context{#1}%
    }%
  }
  \MT@addto@setup{%
    \define@key{MT@pr@c}{context}[]{%
      \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
      \ifx\MT@copy@font\MT@copy@font@\else
        \MT@warning@nl{If protrusion contexts don't work as expected,
          \MessageBreak load the package with the `copyfonts' option}%
      \fi
    }%
  }
⟨/pdf-|lua-⟩
⟨*pdf-⟩
}{
  \define@key{MT@ex@c}{context}[]{%
    \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
        or later. Ignoring `context' key\on@line}%
      {Upgrade pdftex.}%
  }
⟨/pdf-⟩
⟨*pdf-|xe-⟩
  \define@key{MT@pr@c}{context}[]{%
    \MT@error{Protrusion contexts only work with pdftex
⟨pdf-⟩       1.40.4\MessageBreak or later.
⟨xe-⟩       \MessageBreak or luatex.
        Ignoring `context' key\on@line}%
⟨pdf-⟩     {Upgrade pdftex.}%
⟨xe-⟩     {Use pdftex or luatex.}%
  }
⟨/pdf-|xe-⟩
⟨pdf-⟩}
```

\MT@warn@nodim

```
⟨*package⟩
\def\MT@warn@nodim#1{%
  \MT@warning{`\@tempa' is not a dimension.\MessageBreak
              Ignoring it and setting values relative to\MessageBreak #1}%
}
```

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

```
\define@key{MT@pr@c}{unit}[character]{%
  \MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@empty
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{character}\relax{%
```

Test whether it's a dimension, but do not translate it into its final form here, since it may be font-specific.

```
    \MT@ifdimen\@tempa
      {\MT@glet@nc{MT@pr@c@\MT@curr@set@name @unit}\@tempa}%
      {\MT@warn@nodim{character widths}}%
  }%
}
```

Tracking may only be relative to a dimension.

```
\define@key{MT@tr@c}{unit}[1em]{%
  \MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@empty
  \def\@tempa{#1}%
  \MT@ifdimen\@tempa
    {\MT@glet@nc{MT@tr@c@\MT@curr@set@name @unit}\@tempa}%
    {\MT@warn@nodim{1em}%
     \MT@gdef@n{MT@tr@c@\MT@curr@set@name @unit}{1em}}%
}
⟨/package⟩
```

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

```
⟨*pdf-⟩
\MT@map@clist@n{sp,kn}{%
  \define@key{MT@#1@c}{unit}[space]{%
    \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@empty
    \def\@tempa{##1}%
    \MT@ifstreq\@tempa{character}\relax{%
      \MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\m@ne
      \MT@ifstreq\@tempa{space}\relax{%
        \MT@ifdimen\@tempa
          {\MT@glet@nc{MT@#1@c@\MT@curr@set@name @unit}\@tempa}%
          {\MT@warn@nodim{width of space}}%
      }%
    }%
  }%
}
⟨/pdf-⟩
```

The first argument to \SetExpansion accepts some more options.

```
⟨*pdf-|lua-⟩
\MT@map@clist@n{stretch,shrink,step}{%
  \define@key{MT@ex@c}{#1}[]{%
    \MT@ifempty{##1}\relax{%
      \MT@ifint{##1}{%
```

A space terminates the number.

```
        \MT@gdef@n{MT@ex@c@\MT@curr@set@name @#1}{##1 }%
      }{%
        \MT@warning{%
          Value `##1' for option `#1' is not a number.\MessageBreak
          Ignoring it}%
      }%
    }%
  }%
}
\define@key{MT@ex@c}{auto}[true]{%
  \def\@tempa{#1}%
  \csname if\@tempa\endcsname
```

Don’t use autoexpand for pdfTEX version older than 1.20.

```
⟨pdf-⟩   \MT@requires@pdftex4%
⟨lua-⟩   \MT@requires@luatex3\relax
      {\MT@gdef@n{MT@ex@c@\MT@curr@set@name @auto}{autoexpand}}%
⟨pdf-⟩     {\MT@warning{pdftex too old for automatic font expansion}}%
  \else
⟨pdf-⟩   \MT@requires@pdftex4%
⟨*lua-⟩
    \MT@requires@luatex3{%
      \MT@warning{Non-automatic font expansion doesn't work with\MessageBreak
                  luatex}}%
⟨/lua-⟩
      {\MT@glet@nc{MT@ex@c@\MT@curr@set@name @auto}\@empty}%
⟨pdf-⟩     \relax
  \fi
}
```

```
⟨/pdf-|lua-⟩
```

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

```
⟨*pdf-|lua-|xe-⟩
\MT@define@opt@key{tr}{spacing}
\MT@define@opt@key{tr}{outerspacing}
\MT@define@opt@key{tr}{outerkerning}
\MT@define@opt@key{tr}{features}
```

Which ligatures should be disabled?

```
\define@key{MT@tr@c}{noligatures}[]%
  {\MT@xdef@n{MT@tr@c@\MT@curr@set@name @noligatures}{#1}}
\define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
\define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
\define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}
⟨/pdf-|lua-|xe-⟩
```

1.3.7 Character inheritance

\DeclareCharacterInheritance

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

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

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

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

\MT@set@inh@list

No need to create an inheritance list for tracking.

```
\def\MT@set@inh@list#1#2{%
  \MT@ifempty\MT@inh@feat{%
    \MT@map@clist@c\MT@features{\begingroup
      \MT@ifstreq{##1}{tr}\relax{\MT@declare@char@inh{##1}{#1}{#2}}%
    \endgroup}%
  }{%
    \MT@map@clist@c\MT@inh@feat{\begingroup
      \KV@@sp@def\@tempa{##1}%
      \MT@ifempty\@tempa\relax{%
        \edef\@tempa{\csname MT@rbba@\@tempa\endcsname}%
        \MT@ifstreq\@tempa{tr}\relax{%
          \MT@exp@one@n\MT@declare@char@inh{\@tempa}{#1}{#2}}}%
    \endgroup}%
  }%
  \MT@end@catcodes
}
```

The keys for the optional argument.

```
\MT@map@clist@c\MT@features@long{%
  \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}}
\define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}
```

\MT@declare@char@inh The lists cannot be given a name by the user.

```
\def\MT@declare@char@inh#1#2#3{%
  \MT@edef@n{MT@#1@inh@name}%
    {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
  \MT@let@cn\MT@curr@set@name{MT@#1@inh@name}%
  \MT@ifdefined@c@T\MT@extra@inputenc{%
    \MT@xdef@n{MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
⟨debug⟩\MT@dinfo{1}{creating inheritance list `\@nameuse{MT@#1@inh@name}'}%
  \MT@gdef@n{MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
  \def\MT@permutelist{#1@inh}%
  \setkeys{MT@inh}{#2}%
  \MT@permute
}
```

Parse the second argument. \DeclareCharacterInheritance may also be set up for various combinations. We can reuse the key setup from the configuration lists (\Set...).

```
\MT@define@code@key{encoding}{inh}
\MT@define@code@key@family   {inh}
\MT@define@code@key{series}  {inh}
\MT@define@code@key{shape}   {inh}
\MT@define@code@key@size     {inh}
\MT@define@code@key@font     {inh}
```

\MT@inh@do Now parse the third argument, the inheritance lists. We define the commands \MT@inh@⟨*name*⟩@⟨*slot*⟩@, containing the inheriting characters. They will also be translated to slot numbers here, to save some time. The following will be executed only once, namely the first time this inheritance list is encountered (in \MT@set@⟨*feature*⟩@codes).

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

\MT@inh@split Only gather the inheriting characters here. Their codes will actually be set in \MT@set@⟨*feature*⟩@codes.

```
⟨/package⟩
⟨*pdf-|lua-|xe-⟩
\def\MT@inh@split#1=#2=#3\relax{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty \else
    \expandafter\MT@has@inh@prefix\@tempa()\relax\@nil
    \MT@get@slot
⟨pdf-|lua-⟩    \ifnum\MT@char > \m@ne
⟨xe-⟩    \ifx\MT@char\@empty\else
      \let\MT@val\MT@char
      \MT@map@clist@n{#2}{%
        \def\@tempa{##1}%
        \ifx\@tempa\@empty \else
          \MT@get@slot
⟨pdf-|lua-⟩          \ifnum\MT@char > \m@ne
⟨xe-⟩          \ifx\MT@char\@empty\else
            \ifx\MT@inh@prefix\@empty
              \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @\MT@val @}{{\MT@char}}%
            \else
              \MT@exp@cs\MT@xadd{MT@inh@\MT@listname @prefixes}%
                  {{{\MT@val}{\MT@char}\MT@inh@prefix@}}%
            \fi
          \fi
        \fi
      }%
⟨debug⟩\MT@dinfo@nl{2}{children of #1 (\MT@val):
⟨debug⟩    \@nameuse{MT@inh@\MT@listname @\ifx\MT@inh@prefix\@empty\MT@val @\else prefixes\fi}}%
    \fi
  \fi
}
⟨/pdf-|lua-|xe-⟩
```

\MT@inh@prefix
\MT@has@inh@prefix

If the inheriting character is preceded by (⟨*prefix*⟩), where ⟨*prefix*⟩ is one of `l`, `r` or `lr`, this has a special meaning for protrusion. For the other features, we ignore these settings.

```
⟨*package⟩
\def\MT@has@inh@prefix#1(#2)#3#4\@nil{%
  \let\MT@temp\relax
  \ifx\relax#3%
    \def\@tempa{#1#2}%
    \let\MT@inh@prefix\@empty
  \else
    \MT@ifstreq{\MT@feat}{pr}{%
      \MT@ifstreq{#2}{l}{\def\MT@inh@prefix@{{1000}{0}}\@firstoftwo}{%
        \MT@ifstreq{#2}{r}{\def\MT@inh@prefix@{{0}{1000}}\@firstoftwo}{%
          \MT@ifstreq{#2}{lr}{\def\MT@inh@prefix@{{500}{500}}\@firstoftwo}{%
            \MT@warning@nl{`#2' is not a valid prefix in inheritance list%
              \MessageBreak\MT@listname. Ignoring it}%
            \@secondoftwo}}}%
      {\def\@tempa{#3}%
       \def\MT@inh@prefix{#2}%
       \@gobble}%
      {\@firstofone}%
    }{\@firstofone}%
   {\let\MT@char\m@ne
    \let\MT@temp\@gobble
   }%
  \fi
  \MT@temp
}
```

1.3.8 Permutation

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

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

```
\SetProtrusion
   { encoding = U,
     family   = {euroitc,euroitcs} }
   { E = {100,50} }
\SetProtrusion
   { encoding = U,
     family   = {euroitc,euroitcs},
     shape    = it* }
   { E = {100,} }
```

would yield the following assignments:

```
\MT@gdef@n{MT@pr@c@U/euroitc///}{euroitc}
\MT@gdef@n{MT@pr@c@U/euroitcs///}{euroitc}
\MT@gdef@n{MT@pr@c@U/euroitc//it/}{euroitci}
\MT@gdef@n{MT@pr@c@U/euroitcs//it/}{euroitci}
\MT@gdef@n{MT@pr@c@euroitc}{E={100,50}}
\MT@gdef@n{MT@pr@c@euroitci}{E={100,}}
```

```
\def\MT@permute{%
  \let\MT@cnt@encoding\@ne
  \MT@permute@
```

Undefine commands for the next round.

```
  \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
  \MT@glet\MT@tempsize\@undefined
}
\def\MT@permute@{%
  \let\MT@cnt@family\@ne
  \MT@permute@@
  \MT@increment\MT@cnt@encoding
  \MT@ifdefined@n@T{MT@tempencoding\MT@cnt@encoding}%
    \MT@permute@
}
\def\MT@permute@@{%
  \let\MT@cnt@series\@ne
  \MT@permute@@@
  \MT@increment\MT@cnt@family
  \MT@ifdefined@n@T{MT@tempfamily\MT@cnt@family}%
    \MT@permute@@
}
\def\MT@permute@@@{%
  \let\MT@cnt@shape\@ne
  \MT@permute@@@@
  \MT@increment\MT@cnt@series
  \MT@ifdefined@n@T{MT@tempseries\MT@cnt@series}%
    \MT@permute@@@
}
\def\MT@permute@@@@{%
  \MT@permute@@@@@
  \MT@increment\MT@cnt@shape
  \MT@ifdefined@n@T{MT@tempshape\MT@cnt@shape}%
    \MT@permute@@@@
}
```

\MT@permute@@@@@

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

```
\def\MT@permute@@@@@{%
  \MT@permute@define{encoding}%
  \ifMT@document
    \ifx\MT@tempencoding\@empty \else
      \MT@ifdefined@n@TF{T@\MT@tempencoding}\relax
        {\expandafter\expandafter\expandafter\@gobble}%
    \fi
  \fi
  \MT@permute@@@@@@
}
```

\MT@permute@@@@@@

```
\def\MT@permute@@@@@@{%
  \MT@permute@define{family}%
  \MT@permute@define{series}%
  \MT@permute@define{shape}%
  \edef\@tempa{\MT@tempencoding
              /\MT@tempfamily
              /\MT@tempseries
              /\MT@tempshape
              /\MT@ifdefined@c@T\MT@tempsize *}%
```

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

```
  \MT@ifstreq\@tempa{////}\relax{%
    \ifx\MT@tempencoding\@empty
      \MT@warning{%
        You have to specify an encoding for\MessageBreak
        \@nameuse{MT@abbr@\MT@permutelist} list
        `\@nameuse{MT@\MT@permutelist @name}'.\MessageBreak
        Ignoring it}%
    \else
      \MT@ifdefined@c@TF\MT@tempsize{%
```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```
        \MT@ifdefined@n@T{MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}{%
          \MT@map@tlist@c\MT@tempsize\MT@check@rlist
        }%
        \MT@exp@cs\MT@xaddb
          {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
          \MT@tempsize
⟨debug⟩\MT@dinfo@nl{1}{initialising: use list for font \@tempa,\MessageBreak
⟨debug⟩         sizes: \csname MT@\MT@permutelist @\@tempa\MT@extra@context
⟨debug⟩                        @sizes\endcsname}%
      }{%
```

Only one list can apply to a given combination. But we don't warn if the overridden list is to be loaded by the current one.

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

\MT@permute@define

Define the commands.

```
\def\MT@permute@define#1{%
  \@tempcnta=\csname MT@cnt@#1\endcsname\relax
  \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
    {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
    {\MT@let@nc{MT@temp#1}\@empty}%
}
```

\MT@permute@reset

Reset the commands.

```
\def\MT@permute@reset#1{%
  \@tempcnta=\@ne
  \MT@loop
    \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
    \advance\@tempcnta\@ne
    \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
      \iftrue
      \iffalse
  \MT@repeat
}
```

\MT@check@rlist

For every new range item in `\MT@tempsize`, check whether it overlaps with ranges in the existing list.

```
\def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

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

```
\def\MT@check@rlist@#1#2#3{%
  \def\@tempb{#1}%
  \def\@tempc{#2}%
  \MT@if@false
  \MT@exp@cs\MT@map@tlist@c
    {MT@\MT@permutelist @\@tempa\MT@extra@context @sizes}%
    \MT@check@range
}
```

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

```
\def\MT@check@range#1{\expandafter\MT@check@range@ #1}
```

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

```
\def\MT@check@range@#1#2#3{%
  \MT@ifdim{#2}=\m@ne{%
    \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
      \MT@ifdim\@tempb={#1}\MT@if@true\relax
    }{%
```

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

```
      \MT@ifdim\@tempb>{#1}\relax{%
        \MT@ifdim\@tempc>{#1}{%
          \MT@if@true
          \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
        }\relax
      }%
    }%
  }{%
    \MT@ifdim\@tempc=\m@ne{%
```

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

```
      \MT@ifdim\@tempb<{#2}{%
        \MT@ifdim\@tempb<{#1}\relax\MT@if@true
      }\relax
    }{%
```

- Both items are ranges.

```
      \MT@ifdim\@tempb<{#2}{%
        \MT@ifdim\@tempc>{#1}{%
          \MT@if@true
          \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
        }\relax
      }\relax
    }%
  }%
  \ifMT@if@
    \MT@ifstreq{#3}%
        {\csname MT@\MT@permutelist @\csname MT@\MT@permutelist @name\endcsname @load\endcsname}%
        \relax{%
      \MT@warning{\@nameuse{MT@abbr@\MT@permutelist} list
        `\@nameuse{MT@\MT@permutelist @name}' will override\MessageBreak
        list `#3' for font \@tempa,\MessageBreak size \@tempb}%
    }%
```

If we've already found a conflict with this item, we can skip the rest of the list.

```
    \expandafter\MT@tlist@break
```

```
  \fi
}
```

1.4 Package options

1.4.1 Declaring the options

\ifMT@opt@expansion \ifMT@opt@auto \ifMT@opt@DVI

Keep track of whether the user explicitly set these options.

```
\newif\ifMT@opt@expansion
\newif\ifMT@opt@auto
\newif\ifMT@opt@DVI
```

\MT@optwarn@admissible

Some warnings.

```
\def\MT@optwarn@admissible#1#2{%
  \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
                 `#2'. Assuming `false'}%
}
```

\MT@optwarn@nan

```
⟨/package⟩
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\def\MT@optwarn@nan#1#2{%
  \MT@warning@nl{Value `#1' for option `#2' is not a\MessageBreak number.
                 Using default value of \number\@nameuse{MT@#2@default}}%
}
⟨plain⟩}\relax
⟨/package|letterspace⟩
⟨*package⟩
```

\MT@opt@def@set

```
\def\MT@opt@def@set#1{%
  \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
    \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
  }{%
    \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
    \MT@warning@nl{The #1 set `\MT@val' is undeclared.\MessageBreak
                   Using set `\@nameuse{MT@\@tempb @setname}' instead}%
  }%
}
```

expansion and protrusion may be true, false, compatibility, nocompatibility and/or a ⟨*set name*⟩.

```
\MT@map@clist@n{protrusion,expansion}{%
  \define@key{MT}{#1}[true]{%
    \csname MT@opt@#1true\endcsname
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \csname MT@#1true\endcsname
        \edef\@tempb{\csname MT@rbba@#1\endcsname}%
        \MT@ifstreq\MT@val{true}\relax
        {%
          \MT@ifstreq\MT@val{false}{%
            \csname MT@#1false\endcsname
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@let@nc{MT@\@tempb @level}\@ne
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@let@nc{MT@\@tempb @level}\tw@
              }{%
```

If everything failed, it should be a set name.

```
                \MT@opt@def@set{#1}%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}
```

activate is a shortcut for protrusion and expansion.

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

spacing, kerning and tracking do not have a compatibility level.

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

\MT@def@bool@opt

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

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

Boolean options that only set the switch.

```
\MT@map@clist@n{draft,selected,babel}{%
  \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
\MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotrue}
```

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

```
⟨/package⟩
⟨*pdf-|lua-|xe-⟩
⟨lua-⟩\MT@requires@luatex4{\let\pdfoutput\outputmode}\relax
\MT@def@bool@opt{DVIoutput}{%
```

```
  \csname if\@tempa\endcsname
⟨*pdf-|lua-⟩
    \ifnum\pdfoutput>\z@ \MT@opt@DVItrue \fi
    \pdfoutput\z@
  \else
    \ifnum\pdfoutput<\@ne \MT@opt@DVItrue \fi
    \pdfoutput\@ne
⟨/pdf-|lua-⟩
⟨xe-⟩    \MT@warning@nl{Ignoring `DVIoutput' option}%
  \fi
}
⟨/pdf-|lua-|xe-⟩
```

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

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

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

copyfonts will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This options is also *undocumented* in the hope that we can always find out automatically whether it's required. It also works with LuaTEX 0.30 or newer.

```
⟨*pdf-|lua-⟩
⟨pdf-⟩\MT@requires@pdftex7{
  \MT@def@bool@opt{copyfonts}{%
    \csname if\@tempa\endcsname
      \MT@glet\MT@copy@font\MT@copy@font@
    \else
      \MT@glet\MT@copy@font\relax
    \fi
  }
⟨pdf-⟩}{
⟨/pdf-|lua-⟩
⟨*pdf-|xe-⟩
  \MT@def@bool@opt{copyfonts}{%
    \csname if\@tempa\endcsname
      \MT@error
⟨pdf-⟩        {The pdftex version you are using is too old\MessageBreak
⟨pdf-⟩        to use the `copyfonts' option}{Upgrade pdftex.}%
⟨xe-⟩        {The `copyfonts' option does not work with xetex}
⟨xe-⟩        {Use pdftex or luatex instead.}%
    \fi
  }
⟨pdf-⟩}
⟨/pdf-|xe-⟩
```

final is the opposite to draft. It's only kept for backwards compatibility.

```
⟨*package⟩
\MT@def@bool@opt{final}{}
```

The disable option replaces the draft option, which could be inherited from the class options. The third value ifdraft mimicks this behaviour.

```
\define@key{MT}{disable}[true]{%
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{true}\MT@disabletrue{%
    \MT@ifstreq\@tempa{ifdraft}{\ifMT@draft\MT@disabletrue\fi}{%
      \MT@ifstreq\@tempa{false}\relax{%
        \MT@optwarn@admissible{#1}{disable}%
      }%
    }%
  }%
}
```

For verbose output, we redefine \MT@vinfo.

```
\define@key{MT}{verbose}[true]{%
  \let\MT@vinfo\MT@info@nl
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{true}\relax{%
```

Take problems seriously.

```
    \MT@ifstreq\@tempa{errors}{%
      \let\MT@warning   \MT@warn@err
      \let\MT@warning@nl\MT@warn@err
    }{%
      \let\MT@vinfo\@gobble
```

Cast warnings to the winds.

```
      \MT@ifstreq\@tempa{silent}{%
        \let\MT@warning   \MT@info
        \let\MT@warning@nl\MT@info@nl
      }{%
        \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
      }%
    }%
  }%
}
⟨/package⟩
```

Options with numerical keys: factor, stretch, shrink, step, letterspace.

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\MT@map@clist@n{%
⟨package⟩    stretch,shrink,step,%
    letterspace}{%
  \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
    \def\@tempa{##1 }%
```

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

```
    \MT@ifint\@tempa
      {\MT@edef@n{MT@#1}{\@tempa}}%
      {\MT@optwarn@nan{##1}{#1}}%
  }%
}
⟨plain⟩}\relax
⟨/package|letterspace⟩
```

factor will define the protrusion factor only.

```
⟨*package⟩
\define@key{MT}{factor}[\MT@factor@default]{%
  \def\@tempa{#1 }%
  \MT@ifint\@tempa
    {\edef\MT@pr@factor{\@tempa}}
    {\MT@optwarn@nan{#1}{factor}}%
}
```

Unit for protrusion codes.

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

\MT@patches@list
\MT@nopatches@list

The patch and nopatch options. Remember chosen option for later (\relax means 'all', \@empty means 'none').

```
\let\MT@patches@list\relax
\let\MT@nopatches@list\@empty
```

```
\define@key{MT}{patch}[all]{%
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{all}
    \relax
    {\MT@ifstreq\@tempa{none}
      {\let\MT@patches@list\@empty}
      {\def\MT@patches@list{#1}}}%
}
\define@key{MT}{nopatch}[all]{%
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{all}
    {\let\MT@nopatches@list\relax}
    {\MT@ifstreq\@tempa{none}
      \relax
      {\def\MT@nopatches@list{#1}}}%
}
```

We can only apply the patches AtBeginDocument.

```
\MT@addto@setup{%
  \ifx\MT@patches@list\relax
    \let\MT@patches@list\MT@patches@def
  \fi
  \ifx\MT@nopatches@list\@empty\else
    \ifx\MT@nopatches@list\relax
      \let\MT@nopatches@list\MT@patches@def
    \fi
    \MT@map@clist@c\MT@nopatches@list{%
      \MT@rem@from@clist{#1}\MT@patches@list}%
  \fi
  \ifx\MT@patches@list\@empty\else
^^X    \MT@map@clist@c\MT@patches@list{\MT@apply@patch{#1}}%
^^Q    \MT@warning@nl{Patches require the etex extensions. Ignoring them}%
  \fi
}
```

1.4.2 Loading the definition file

Load the engine-specific code (as strewn across this file).

\MT@get@MT@version
\MT@version
\MT@check@MT@version

We also check whether versions are the same.

```
\def\MT@get@MT@version#1 #2 #3\@nil{#1 #2}
\edef\MT@version{\expandafter\expandafter\expandafter\MT@get@MT@version
  \csname ver@\MT@MT.sty\endcsname\@nil}
\def\MT@check@MT@version#1#2{%
  \MT@ifstreq\MT@version{#1}{}{%
    \MT@warning@nl{Mismatching file versions:\MessageBreak
       \MT@MT.sty provides:\MessageBreak`\MT@version',\MessageBreak
       whereas #2 provides:\MessageBreak`#1'.\MessageBreak
       Please fix your installation}}}
\input{\MT@MT-\MT@engine tex.def}
\edef\@tempa{\expandafter\expandafter\expandafter\MT@get@MT@version
  \csname ver@\MT@MT-\MT@engine tex.def\endcsname\@nil}
\MT@check@MT@version\@tempa{\MT@MT-\MT@engine tex.def}
```

1.4.3 Reading the configuration file

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

```
\MT@protrusiontrue
⟨/package⟩
⟨*pdf-|lua-⟩
\ifnum\pdfoutput<\@ne \else
```

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

```
⟨pdf-⟩  \MT@requires@pdftex4{
    \MT@expansiontrue
⟨pdf-⟩     \MT@autotrue
⟨pdf-⟩  }\relax
\fi
⟨lua-⟩\MT@autotrue
⟨/pdf-|lua-⟩
```

\MT@config@file

The main configuration file will be loaded before processing the package options. However, the config option must of course be evaluated beforehand. We also have to define a no-op for the regular option processing later.

```
⟨*package⟩
\define@key{MT}{config}[]{\relax}
\def\MT@temp#1config=#2,#3\@nil{%
  \MT@ifempty{#2}%
    {\def\MT@config@file{\MT@MT.cfg}}%
    {\def\MT@config@file{#2.cfg}}%
}
\expandafter\expandafter\expandafter\MT@temp
  \csname opt@\@currname.\@currext\endcsname,config=,\@nil
```

Load the file.

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

\MT@check@active@set

We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by \DeclareMicrotypeSetDefault (this is done at the end of the preamble).

```
\def\MT@check@active@set#1{%
```

```
  \MT@ifdefined@n@TF{MT@#1@setname}{%
    \MT@info@nl{Using \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
  }{%
    \MT@ifdefined@n@TF{MT@default@#1@set}{%
      \MT@glet@nn{MT@#1@setname}{MT@default@#1@set}%
      \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set `\@nameuse{MT@#1@setname}'}%
    }{%
```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '`@`', and issue a warning.

```
      \MT@gdef@n{MT@#1@setname}{@}%
      \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
                     \MessageBreak Using empty set}%
    }%
  }%
}
```

1.4.4 Hook for other packages

\Microtype@Hook

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

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

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

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

`\MicroType@Hook` with a capital `T` (which only existed in version 1.7) is now officially deprecated.

```
\MT@ifdefined@c@T\MicroType@Hook{\MT@error{%
  Command \@backslashchar MicroType@Hook is deprecated.\MessageBreak
  Use \@backslashchar Microtype@Hook instead}
  {You might want to inform the font package authors.}\MicroType@Hook}
\MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

1.4.5 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` (but specifying font sets is not allowed), and `patch` and `nopatch`.

```
\DeclareRobustCommand\microtypesetup{\setkeys{MT}}
\MT@addto@setup{\DeclareRobustCommand\microtypesetup[1]{\setkeys{MTX}{#1}\selectfont}}
⟨/package⟩
```

```
⟨*pdf-|lua-|xe-⟩
\def\MT@define@optionX#1#2{%
  \define@key{MTX}{#1}[true]{%
    \edef\@tempb{\csname MT@rbba@#1\endcsname}%
    \MT@map@clist@n{##1}{%
      \KV@@sp@def\MT@val{####1}%
      \MT@ifempty\MT@val\relax{%
        \@tempcnta=\m@ne
        \MT@ifstreq\MT@val{true}{%
```

Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

```
          \MT@checksetup{#1}{%
            \@tempcnta=\csname MT@\@tempb @level\endcsname
            \MT@vinfo{Enabling #1
                    (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
          }%
        }{%
          \MT@ifstreq\MT@val{false}{%
            \@tempcnta=\z@
            \MT@vinfo{Disabling #1\on@line}%
          }{%
            \MT@ifstreq\MT@val{compatibility}{%
              \MT@checksetup{#1}{%
                \@tempcnta=\@ne
                \MT@let@nc{MT@\@tempb @level}\@ne
                \MT@vinfo{Setting #1 to level 1\on@line}%
              }%
            }{%
              \MT@ifstreq\MT@val{nocompatibility}{%
                \MT@checksetup{#1}{%
                  \@tempcnta=\tw@
                  \MT@let@nc{MT@\@tempb @level}\tw@
                  \MT@vinfo{Setting #1 to level 2\on@line}%
                }%
              }{\MT@error{Value `\MT@val' for key `#1' not recognised}
                         {Use any of `true', `false', `compatibility' or
                          `nocompatibility'.}%
              }%
            }%
          }%
        }%
        \ifnum\@tempcnta>\m@ne
          #2\@tempcnta\relax
        \fi
      }%
    }%
  }%
}
```

\MT@checksetup

Test whether the feature wasn't disabled in the package options.

```
\def\MT@checksetup#1{%
  \csname ifMT@#1\endcsname
    \expandafter\@firstofone
  \else
    \MT@error{You cannot enable #1 if it was disabled\MessageBreak
              in the package options}{Load microtype with #1 enabled.}%
    \expandafter\@gobble
  \fi
}
```

```
\MT@define@optionX{protrusion}\MT@protrudechars
⟨*pdf-|lua-⟩
\MT@define@optionX{expansion}\MT@adjustspacing
```

\MT@protrudechars

\MT@adjustspacing

```
⟨*lua-⟩
\MT@requires@luatex4{
  \let\pdfprotrudechars\protrudechars
  \let\pdfadjustspacing\adjustspacing
}\relax
⟨/lua-⟩
\let\MT@protrudechars\pdfprotrudechars
\let\MT@adjustspacing\pdfadjustspacing
⟨/pdf-|lua-⟩
⟨*xe-⟩
\let\MT@protrudechars\XeTeXprotrudechars
\define@key{MTX}{expansion}[true]{\MT@warning{Ignoring expansion setup}}
⟨/xe-⟩
```

\MT@define@optionX@

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

```
⟨pdf-⟩\MT@requires@pdftex6{
⟨lua-⟩\MT@requires@luatex3{
  \def\MT@define@optionX@#1#2{%
    \define@key{MTX}{#1}[true]{%
      \MT@map@clist@n{##1}{%
        \KV@@sp@def\MT@val{####1}%
        \MT@ifempty\MT@val\relax{%
          \@tempcnta=\m@ne
          \MT@ifstreq\MT@val{true}{%
            \MT@checksetup{#1}{%
              \@tempcnta=\@ne
              \MT@vinfo{Enabling #1\on@line}%
            }%
          }{%
            \MT@ifstreq\MT@val{false}{%
              \@tempcnta=\z@
              \MT@vinfo{Disabling #1\on@line}%
            }{\MT@error{Value `\MT@val' for key `#1' not recognised}
                       {Use either `true' or `false'}%
            }%
          }%
          \ifnum\@tempcnta>\m@ne
            #2\relax
          \fi
        }%
      }%
    }%
  }
```

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

```
  \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
                               \else \let\MT@tracking\MT@tracking@ \fi}
⟨pdf-⟩  \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
⟨pdf-⟩  \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
⟨pdf-⟩                               \pdfappendkern\@tempcnta}
⟨pdf-|lua-⟩}{
```

Disable for older pdfTeX versions and for XƎTeX and LuaTeX.

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

```
}
⟨/pdf-|lua-|xe-⟩
```

\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. These options are *undocumented*, as they completely deprive us of the possibility to act – we're blind and paralysed.

```
⟨*package⟩
\let\MT@saved@setupfont\MT@setupfont

\define@key{MTX}{deactivate}[]{%
  \MT@info{Deactivate `\MT@MT' package}%
  \let\MT@setupfont\relax
}
\define@key{MTX}{reactivate}[]{%
  \MT@info{Reactivate `\MT@MT' package}%
  \let\MT@setupfont\MT@saved@setupfont
}
```

Apply or revert patches.

```
\define@key{MTX}{patch}[all]{%
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{all}
    {\let\@tempa\MT@patches@def}
    {\MT@ifstreq\@tempa{none}
      {\let\@tempa\@empty}
      \relax}%
  \ifx\@tempa\@empty\else
^^X    \MT@map@clist@c\@tempa{\MT@apply@patch{##1}}%
^^Q    \MT@warning@nl{Patches require the etex extensions. Ignoring them}%
  \fi
}
\define@key{MTX}{nopatch}[all]{%
  \def\@tempa{#1}%
  \MT@ifstreq\@tempa{all}
    {\let\@tempa\MT@patches@def}
    {\MT@ifstreq\@tempa{none}
      {\let\@tempa\@empty}
      \relax}%
  \ifx\@tempa\@empty\else
^^X    \MT@map@clist@c\@tempa{\MT@undo@patch{##1}}%
  \fi
}
⟨/package⟩
```

1.4.6 Processing the options

\MT@ProcessOptionsWithKV

Parse options.

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\def\MT@ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\MT@temp\@empty
⟨plain⟩  \MT@requires@latex2{
    \MT@map@clist@c\@classoptionslist{%
      \def\CurrentOption{##1}%
      \MT@ifdefined@n@T{KV@#1@\expandafter\MT@getkey\CurrentOption=\@nil}{%
        \edef\MT@temp{\MT@temp,\CurrentOption,}%
        \@expandtwoargs\@removeelement\CurrentOption
          \@unusedoptionlist\@unusedoptionlist
      }%
    }%
    \edef\MT@temp{\noexpand\setkeys{#1}%
```

```
                    {\MT@temp\@ptionlist{\@currname.\@currext}}}%
```

eplain can handle package options.

```
⟨*plain⟩
  }{\edef\MT@temp{\noexpand\setkeys{#1}%
                    {\csname usepkg@options@\usepkg@pkg\endcsname}}}
⟨/plain⟩
  \MT@temp
  \MT@clear@options
}
```

\MT@getkey

For key=val in class options.

```
\def\MT@getkey#1=#2\@nil{#1}
```

```
\MT@ProcessOptionsWithKV{MT}
⟨plain⟩}\relax
⟨/package|letterspace⟩
⟨*package⟩
```

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

```
\MT@addto@setup{%
\ifMT@disable
```

We disable most of what we've just defined in the 5714 lines above if we are running in disable (aka. draft) mode.

```
  \MT@warning@nl{The `disable' option is in effect.\MessageBreak
                 Disabling all micro-typographic extensions.\MessageBreak
                 This might lead to different line and page breaks}%
  \let\MT@setupfont\relax
  \renewcommand*\LoadMicrotypeFile[1]{}%
  \renewcommand*\microtypesetup[1]{}%
  \renewcommand*\microtypecontext[1]{}%
  \renewcommand*\lsstyle{}%
\else
  \MT@setup@PDF
  \MT@setup@copies
```

Fix the font sets.

```
  \MT@map@tlist@c\MT@font@sets\MT@fix@font@set
  \MT@setup@protrusion
  \MT@setup@expansion
  \MT@setup@tracking
  \MT@setup@warntracking
  \MT@setup@spacing
  \MT@setup@kerning
  \MT@setup@noligatures
}
⟨/package⟩
```

\MT@setup@PDF

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

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

```
⟨*pdf-|lua-⟩
\def\MT@setup@PDF{%
  \MT@info@nl{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
              \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%
}
```

\MT@setup@copies

Working on font copies?

```
\def\MT@setup@copies{%
  \ifx\MT@copy@font\relax\else \MT@info@nl{Using font copies for contexts}\fi
}
⟨/pdf-|lua-⟩
⟨*xe-⟩
\let\MT@setup@PDF\relax
\let\MT@setup@copies\relax
⟨/xe-⟩
```

\MT@setup@protrusion

Protrusion.

```
⟨*pdf-|lua-|xe-⟩
\def\MT@setup@protrusion{%
  \ifMT@protrusion
    \edef\MT@active@features{\MT@active@features,pr}%
    \MT@protrudechars\MT@pr@level
    \MT@info@nl{Character protrusion enabled (level \number\MT@pr@level)%
      \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
        factor: \number\MT@pr@factor\fi
      \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}%
    \MT@check@active@set{pr}%
  \else
    \let\MT@protrusion\relax
    \MT@info@nl{No character protrusion}%
  \fi
}
⟨/pdf-|lua-|xe-⟩
```

\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.

```
⟨*pdf-|lua-⟩
\def\MT@setup@expansion{%
  \ifnum\pdfoutput<\@ne
    \ifMT@opt@expansion
⟨*lua-⟩
      \ifMT@expansion
        \MT@requires@luatex3{%
          \MT@warning@nl{Font expansion doesn't work properly with luatex in\MessageBreak
             DVI mode: the glyphs won't be actually transformed,\MessageBreak
             but will only be shifted. You might want to use\MessageBreak
             pdflatex instead. I'll continue anyway ..}%
          %\MT@expansionfalse
        }\relax
      \fi
⟨/lua-⟩
    \else
      \MT@expansionfalse
    \fi
  \fi
  \ifMT@expansion
```

Set up the values for font expansion: if stretch has not been specified, we take the default value of 20.

```
    \ifnum\MT@stretch=\m@ne
      \let\MT@stretch\MT@stretch@default
    \fi
```

If shrink has not been specified, it will inherit the value from stretch.

```
    \ifnum\MT@shrink=\m@ne
      \let\MT@shrink\MT@stretch
    \fi
```

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

```
    \ifnum\MT@step=\m@ne
⟨pdf-⟩    \MT@requires@pdftex6{%
      \def\MT@step{1 }%
⟨*pdf-⟩
    }{%
      \ifnum\MT@stretch>\MT@shrink
        \ifnum\MT@shrink=\z@
          \@tempcnta=\MT@stretch
        \else
          \@tempcnta=\MT@shrink
        \fi
      \else
        \ifnum\MT@stretch=\z@
          \@tempcnta=\MT@shrink
        \else
          \@tempcnta=\MT@stretch
        \fi
      \fi
      \divide\@tempcnta 5\relax
      \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
      \edef\MT@step{\number\@tempcnta\space}%
    }%
⟨/pdf-⟩
    \fi
    \ifnum\MT@step=\z@
      \MT@warning@nl{The expansion step cannot be set to zero.\MessageBreak
          Setting it to one}%
      \def\MT@step{1 }%
    \fi
```

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

```
    \let\MT@auto\@empty
    \ifMT@auto
```

We turn off automatic expansion if output mode is DVI.

```
⟨*pdf-⟩
      \MT@requires@pdftex4{%
        \ifnum\pdfoutput<\@ne
          \ifMT@opt@auto
            \MT@error{%
              Automatic font expansion only works for PDF output.\MessageBreak
              However, you are creating a DVI file}
             {If you have created expanded fonts instances, remove `auto' from%
              \MessageBreak the package options. Otherwise, you have to switch
              off expansion\MessageBreak completely.}%
          \fi
          \MT@autofalse
        \else
          \def\MT@auto{autoexpand}%
        \fi
```

Also, if pdfTEX is too old.

```
      }{%
        \MT@error{%
          The pdftex version you are using is too old for\MessageBreak
          automatic font expansion}%
         {If you have created expanded fonts instances, remove `auto' from\MessageBreak
          the package options. Otherwise, you have to switch off expansion\MessageBreak
          completely, or upgrade pdftex to version 1.20 or newer.}%
        \MT@autofalse
        \def\MT@auto{1000 }%
      }%
⟨/pdf-⟩
⟨lua-⟩      \MT@requires@luatex3\relax{\def\MT@auto{autoexpand}}%
    \else
⟨*pdf-⟩
```

No automatic expansion.

```
      \MT@requires@pdftex4\relax{%
        \def\MT@auto{1000 }%
      }%
⟨/pdf-⟩
⟨*lua-⟩
      \MT@requires@luatex3{%
        \ifMT@opt@auto
          \MT@error{Non-automatic font expansion does not work with\MessageBreak
                    luatex}{Remove `auto=false' from the package options, or use pdftex.}%
          \MT@autotrue
        \fi
      }\relax
⟨/lua-⟩
    \fi
```

Choose the appropriate macro for selected expansion.

```
    \ifMT@selected
      \let\MT@set@ex@codes\MT@set@ex@codes@s
    \else
      \let\MT@set@ex@codes\MT@set@ex@codes@n
    \fi
```

Filter out `stretch=0,shrink=0`, since it would result in a pdfTEX error.

```
    \ifnum\MT@stretch=\z@
      \ifnum\MT@shrink=\z@
        \MT@warning@nl{%
          Both the stretch and shrink limit are set to zero.\MessageBreak
          Disabling font expansion}%
        \MT@expansionfalse
      \fi
    \fi
  \fi
  \ifMT@expansion
    \edef\MT@active@features{\MT@active@features,ex}%
    \MT@adjustspacing\MT@ex@level
    \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
                (level \number\MT@ex@level),\MessageBreak
                stretch: \number\MT@stretch, shrink: \number\MT@shrink,
                step: \number\MT@step, \ifMT@selected\else non-\fi selected}%
```

\MT@check@step

Check whether `stretch` and `shrink` are multiples of `step`.

```
    \def\MT@check@step##1{%
      \@tempcnta=\csname MT@##1\endcsname
      \divide\@tempcnta \MT@step
      \multiply\@tempcnta \MT@step
      \ifnum\@tempcnta=\csname MT@##1\endcsname\else
        \MT@warning@nl{The ##1 amount is not a multiple of step.\MessageBreak
                       The effective maximum ##1 is \the\@tempcnta\space
                       (step \number\MT@step)}%
      \fi
    }%
    \MT@check@step{stretch}%
    \MT@check@step{shrink}%
    \MT@check@active@set{ex}%
```

\showhyphens

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

```
    \MT@ifdefined@n@TF{showhyphens }{%
      \def\MT@temp##1##2{%
        \MT@exp@cs\CheckCommand{showhyphens }[1]{##1}%
        \DeclareRobustCommand\showhyphens[1]{##2}}%
    }{%
      \def\MT@temp##1##2{%
        \CheckCommand*\showhyphens[1]{##1}%
        \gdef\showhyphens####1{##2}}%
    }%
    \MT@temp
       {\setbox0\vbox{\color@begingroup
        \everypar{}\parfillskip\z@skip
        \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
        \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}
       {\setbox0\vbox{\color@begingroup\pdfadjustspacing\z@
        \everypar{}\parfillskip\z@skip
        \hsize\maxdimen\normalfont\pretolerance\m@ne\tolerance\m@ne
        \hbadness\z@\showboxdepth\z@\ ##1\color@endgroup}}%
```

```
  \else
    \let\MT@expansion\relax
    \MT@info@nl{No font expansion}%
  \fi
}
⟨/pdf-|lua-⟩
⟨*xe-⟩
\def\MT@setup@expansion{%
  \ifMT@expansion
    \ifMT@opt@expansion
      \MT@error{Font expansion does not work with xetex}
               {Use pdftex or luatex instead.}%
    \fi
  \fi
}
⟨/xe-⟩
```

\MT@setup@tracking

Tracking, spacing and kerning.

```
⟨*pdf-|lua-|xe-⟩
⟨pdf-⟩\MT@requires@pdftex6{%
⟨lua-⟩\MT@requires@luatex3{%
  \def\MT@setup@tracking{%
    \ifMT@tracking
      \MT@info@nl{Tracking enabled}%
      \MT@check@active@set{tr}%
```

Enable protrusion for compensation at the line edges.

```
      \ifMT@protrusion\else\MT@protrudechars\@ne\fi
    \else
      \let\MT@tracking\relax
      \MT@info@nl{No adjustment of tracking}%
    \fi
  }
⟨/pdf-|lua-|xe-⟩
```

\MT@setup@spacing

```
⟨*pdf-⟩
  \def\MT@setup@spacing{%
    \ifMT@spacing
      \edef\MT@active@features{\MT@active@features,sp}%
      \pdfadjustinterwordglue\@ne
      \MT@info@nl{Adjustment of interword spacing enabled}%
```

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

```
      \MT@with@package@T{ragged2e}{%
        \MT@warning@nl{You are using the `ragged2e' package.\MessageBreak
          Adjustment of interword spacing may lead to\MessageBreak
          undesired results when used with `ragged2e'.\MessageBreak
          In this case, disable the `spacing' option}%
      }%
      \MT@check@active@set{sp}%
    \else
      \let\MT@spacing\relax
      \MT@info@nl{No adjustment of interword spacing}%
    \fi
  }
```

\MT@setup@spacing@check

Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue > 0. Why 1500? Because some packages redefine \frenchspacing.[9]

```
  \def\MT@setup@spacing@check{%
    \ifMT@spacing
      \ifMT@babel \else
        \ifnum\sfcode`\. > 1500
          \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
            \MT@warning@nl{%
              \@backslashchar nonfrenchspacing is active. Adjustment of\MessageBreak
              interword spacing will disable it. You might want\MessageBreak
              to add `\@backslashchar microtypecontext{spacing=nonfrench}'\MessageBreak
              to your preamble}%
          }%
        \fi
      \fi
    \fi
  }
```

\MT@setup@kerning

```
  \def\MT@setup@kerning{%
    \ifMT@kerning
      \edef\MT@active@features{\MT@active@features,kn}%
      \pdfprependkern\@ne
      \pdfappendkern\@ne
      \MT@info@nl{Adjustment of character kerning enabled}%
      \MT@check@active@set{kn}%
    \else
      \let\MT@kerning\relax
      \MT@info@nl{No adjustment of character kerning}%
    \fi
  }
⟨/pdf-⟩
```

\MT@error@doesnt@work

If pdfTEX is too old, we disable tracking, spacing and kerning, and throw an error message. We also switch the features off for LuaTEX and XƎTEX.

```
⟨pdf-|lua-⟩}{
⟨*lua-⟩
  \def\MT@setup@tracking{%
```

[9] Cf. the c.t.t. thread '\frenchspacing with AMS packages and babel', started by Philipp Lehman on 16 August 2005, MID: ddtbaj$rob$1@online.de

```
    \ifMT@tracking
      \MT@error{The tracking feature only works with luatex 0.62\MessageBreak
        or newer. Switching it off}{Upgrade luatex.}%
      \MT@trackingfalse
      \MT@let@nc{MT@tracking}\relax
    \else
      \MT@info@nl{No adjustment of tracking (luatex too old)}%
    \fi
  }
}
⟨/lua-⟩
⟨*pdf-|lua-|xe-⟩
  \def\MT@error@doesnt@work#1{%
    \csname ifMT@#1\endcsname
      \MT@error{The #1 feature only works with pdftex 1.40\MessageBreak
        or newer. Switching it off}
⟨pdf-⟩        {Upgrade pdftex.}%
⟨lua-|xe-⟩        {Use pdftex instead.}%
      \csname MT@#1false\endcsname
      \MT@let@nc{MT@#1}\relax
    \else
      \MT@info@nl{No adjustment of #1%
⟨pdf-⟩      \space(pdftex too old)%
      }%
    \fi
  }
⟨pdf-⟩  \def\MT@setup@tracking{\MT@error@doesnt@work{tracking}}
  \def\MT@setup@kerning {\MT@error@doesnt@work{kerning}}
  \def\MT@setup@spacing {\MT@error@doesnt@work{spacing}}
⟨pdf-⟩}
⟨/pdf-|lua-|xe-⟩
```

\MT@setup@warntracking

```
⟨pdf-|lua-|xe-⟩\def\MT@setup@warntracking
⟨letterspace⟩\MT@addto@setup
```

\MT@warn@tracking@DVI

With pdfTEX, we issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```
⟨*pdf-|lua-|xe-|letterspace⟩
{%
⟨*pdf-|letterspace⟩
  \ifnum\pdfoutput<\@ne
    \def\MT@warn@tracking@DVI{%
⟨letterspace⟩      \MT@pdf@or@lua{%
      \MT@warning@nl{%
          You are using tracking/letterspacing in DVI mode.\MessageBreak
          This will probably not work, unless the post-\MessageBreak
          processing program (dvips, dvipdfm(x), ...) is\MessageBreak
          able to create the virtual fonts on the fly}%
⟨letterspace⟩      }\relax
      \MT@glet\MT@warn@tracking@DVI\relax
    }%
  \else
⟨/pdf-|letterspace⟩
⟨*pdf-|lua-|letterspace⟩
    \def\MT@warn@tracking@DVI{%
      \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
      \MT@glet\MT@warn@tracking@DVI\relax
    }%
⟨/pdf-|lua-|letterspace⟩
⟨pdf-|letterspace⟩  \fi

  \ifnum\MT@letterspace=\m@ne
    \let\MT@letterspace\MT@letterspace@default
  \else
    \MT@ls@too@large\MT@letterspace
  \fi
}
⟨/pdf-|lua-|xe-|letterspace⟩
```

\MT@setup@noligatures

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

```
⟨*pdf-|lua-⟩
\def\MT@setup@noligatures{%
⟨pdf-⟩  \MT@requires@pdftex5{%
    \ifMT@noligatures \else
      \let\MT@noligatures\relax
    \fi
⟨pdf-⟩  }\relax
}
⟨/pdf-|lua-⟩
⟨xe-⟩\let\MT@setup@noligatures\relax
```

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

```
⟨*package⟩
\MT@addto@setup{%
  \ifx\MT@active@features\@empty \else
    \edef\MT@active@features{\expandafter\@gobble\MT@active@features}%
  \fi
  \MT@documenttrue
}
```

\MT@set@babel@context

Interaction with babel.

```
\def\MT@set@babel@context#1{%
  \MT@ifdefined@n@TF{MT@babel@#1}{%
    \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
    \expandafter\MT@exp@one@n\expandafter\microtypecontext
      \csname MT@babel@#1\endcsname
  }{%
    \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
  }%
}
```

\MT@shorthandoff

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

```
\@ifpackageloaded{babel}{
  \def\MT@shorthandoff#1#2{%
    \MT@info@nl{Switching off #1 babel's active characters (#2)}%
    \shorthandoff{#2}}
}{
  \def\MT@shorthandoff#1#2{%
    \MT@error{You must load `babel' before `\MT@MT'}
             {Otherwise, `\MT@MT' cannot switch off #1 babel's\MessageBreak
              active characters.}}
}
```

We patch babel's language switching commands to enable language-dependent setup.

```
\MT@addto@setup{%
  \ifMT@babel
    \@ifpackageloaded{babel}{%
      \MT@info@nl{Redefining babel's language switching commands}%
      \let\MT@orig@select@language\select@language
      \def\select@language#1{%
        \MT@orig@select@language{#1}%
        \MT@set@babel@context{#1}%
      }%
      \let\MT@orig@foreign@language\foreign@language
      \def\foreign@language#1{%
        \MT@orig@foreign@language{#1}%
        \MT@set@babel@context{#1}%
      }%
      \ifMT@kerning
```

Disable French babel's active characters.

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

Disable Turkish babel's active characters.

```
        \MT@if@false
        \MT@with@babel@and@T{turkish} \MT@if@true
        \ifMT@if@\MT@shorthandoff{Turkish}{:!=}\fi
      \fi
```

In case babel was loaded before microtype:

```
      \MT@set@babel@context\languagename
```

The polyglossia package has a useful hook. Unfortunately, compatibility with polyglossia is less useful in itself, as only LuaTEX allows working on font copies, and currently doesn't provide the kerning or spacing feature. But who knows, maybe somebody would want more protrusion in French...

```
    }{%
      \@ifpackageloaded{polyglossia}{%
        \MT@info@nl{Registering with polyglossia's language switching hook}%
        \gappto\polyglossia@language@switched{%
          \MT@set@babel@context{\languagename}%
        }%
        \MT@set@babel@context\languagename
      }{%
        \MT@warning@nl{%
          You did not load the babel or the polyglossia package.\MessageBreak
          The `babel' option won't have any effect}%
      }%
    }%
  \fi
}
```

Now we close the \fi from \ifMT@disable.

```
\MT@addto@setup{\fi
```

Set up the current font, most likely the normal font. This has to come after all of the setup (including anything from the preamble) has been dealt with.

```
  \selectfont}
```

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

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

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```
⟨*package|letterspace⟩
⟨plain⟩\MT@requires@latex1{
\AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@\@empty}
⟨plain⟩}\relax
⟨/package|letterspace⟩
```

Must come at the very, very end.

```
⟨package⟩\MT@ifdefined@c@T\MT@setup@spacing@check
⟨package⟩  {\AtBeginDocument{\MT@setup@spacing@check}}
```

Restore catcodes.

```
⟨package|letterspace⟩\MT@restore@catcodes
```

That was that.

2 Configuration files

Let’s now write the font configuration files.

```
⟨*config⟩

```

2.1 Font sets

We first declare some sets in the main configuration file.

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

\DeclareMicrotypeSet{all}
   { }

\DeclareMicrotypeSet{allmath}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U} }

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

\DeclareMicrotypeSet{allmath-nott}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,EU1,EU2,TU,TS1,OML,OMS,U},
     family   = {rm*,sf*}
   }

\DeclareMicrotypeSet{alltext-nott}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     family   = {rm*,sf*}
   }

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

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

\DeclareMicrotypeSet{smallcaps}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     shape    = {sc*,si,scit}
   }

\DeclareMicrotypeSet{footnotesize}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     size     = {-small}
   }

\DeclareMicrotypeSet{scriptsize}
   { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1,EU1,EU2,TU},
     size     = {-footnotesize}
   }

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

```

The default sets.

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

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

```

2.2 Font variants and aliases

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals): Additionally, we add the now common variants for Lining, Tabular, Oldstyle, and Tabular Oldstyle numbers.

```
%%% ----------------------------------------------------------------------
%%% FONT VARIANTS AND ALIASES

\DeclareMicrotypeVariants{x,j,w,a,d,0,1,-LF,-TLF,-OsF,-TOsF}
```

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

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

```

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

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

```
\DeclareMicrotypeAlias{lmsy}{cmsy}         %   "
\DeclareMicrotypeAlias{lmm} {cmm}          %   "
\DeclareMicrotypeAlias{aer} {cmr}          % ae
\DeclareMicrotypeAlias{zer} {cmr}          % zefonts
```

```
\DeclareMicrotypeAlias{cmor}{cmr}          % eco
\DeclareMicrotypeAlias{hfor}{cmr}          % hfoldsty
\DeclareMicrotypeAlias{mlmr}{cmr}          % mlmodern
\DeclareMicrotypeAlias{mlmsy}{cmsy}        %   "
\DeclareMicrotypeAlias{mlmm} {cmm}         %   "
```

Another, new Computer Modern extension. The newcomputermodern package loads it by file name.

```
\DeclareMicrotypeAlias{NewCM10-Book.otf}   {New Computer Modern}
\DeclareMicrotypeAlias{NewCM10-Regular.otf}{New Computer Modern}
```

CMU Serif can use the settings from New Computer Modern too.

```
\DeclareMicrotypeAlias{CMU Serif}          {New Computer Modern}
```

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

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

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

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

The newpx package, a replacement for pxfonts.

```
\DeclareMicrotypeAlias{zpllf}{pplx}        % newpxtext
\DeclareMicrotypeAlias{zplosf}{pplj}       %    "
\DeclareMicrotypeAlias{zpltlf}{pplx}       %    "
\DeclareMicrotypeAlias{zpltosf}{pplj}      %    "
```

The domitian package.

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

The OpenType versions:

```
\DeclareMicrotypeAlias{Palatino Linotype}{Palatino}
\DeclareMicrotypeAlias{Palatino LT Std}  {Palatino}
\DeclareMicrotypeAlias{TeX Gyre Pagella} {Palatino}
\DeclareMicrotypeAlias{Domitian}         {Palatino}
\DeclareMicrotypeAlias{Asana Math}       {Palatino}
%% -- Times New Roman
\DeclareMicrotypeAlias{txr}{ptm}           % txfonts
```

The newtx package, a replacement for txfonts.

```
\DeclareMicrotypeAlias{ntxlf}  {ptmx}      % newtxtext
\DeclareMicrotypeAlias{ntxtlf} {ptmx}      %    "
\DeclareMicrotypeAlias{ntxosf} {ptmj}      %    "
\DeclareMicrotypeAlias{ntxtosf}{ptmj}      %    "
```

The tempora package.

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

The step package.

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

The stix, stix2 and stickstoo packages (the latter two have departed a bit from being a Times clone, but still seem close enough).

```
\DeclareMicrotypeAlias{stix} {ptm}         % stix
\DeclareMicrotypeAlias{stix2}{ptm}         % stix2
\DeclareMicrotypeAlias{SticksTooText-LF}  {ptmx}
\DeclareMicrotypeAlias{SticksTooText-TLF} {ptmx}
```

```
\DeclareMicrotypeAlias{SticksTooText-OsF} {ptmj}
\DeclareMicrotypeAlias{SticksTooText-TOsF}{ptmj}
```

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

MicroPress's Charter version (chmath).

```
%% -- Charter
\DeclareMicrotypeAlias{chr}{bch}           % CH Math
```

The XCharter package extends the Charter fonts.

```
\DeclareMicrotypeAlias{XCharter-TLF} {bch} % XCharter
\DeclareMicrotypeAlias{XCharter-TOsF}{bch} %    "
```

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

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

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

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

Because a configuration file for Adobe Garamond wouldn't be permitted for TEX Live distribution, we use EB Garamond as the base font.

```
\DeclareMicrotypeAlias{pad} {EBGaramond-LF}% Adobe Garamond
\DeclareMicrotypeAlias{padx}{EBGaramond-TLF}%   "
\DeclareMicrotypeAlias{padj}{EBGaramond-TOsF}%  "
%% --
```

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
```

The Lato and Fontin fonts (and many, many more...) only contain a basic set of glyphs. We alias them here to the basic settings (see 3.1.5) to prevent lots of warning messages from the inheritance settings; they will still receive protrusion settings from the default (T1) configuration.

```
\DeclareMicrotypeAlias{Lato}             {TU-basic}
\DeclareMicrotypeAlias{Lato-Regular}     {TU-basic}
\DeclareMicrotypeAlias{Fontin}           {TU-basic}
\DeclareMicrotypeAlias{Fontin-Regular}   {TU-basic}
\DeclareMicrotypeAlias{Bergamo Std}      {TU-basic}
```

The fontawesome and fontawesome5 packages are aliased to empty settings (see 3.1.6 and 3.2.6).

```
\DeclareMicrotypeAlias{FontAwesome}      {TU-empty} % fontawesome
\DeclareMicrotypeAlias{fontawesomefree}  {TU-empty} % fontawesome5
\DeclareMicrotypeAlias{fontawesomepro}   {TU-empty}
\DeclareMicrotypeAlias{fontawesomebrands}{TU-empty}
```

2.3 Interaction with babel

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

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

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

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

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

```

2.4 Note on admissible characters

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

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

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

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

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

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

With XeTeX or LuaTeX, in contrast, it is advisable to use the proper Unicode characters, or the font-specific glyph names prefixed with ‘/’ (cf. section 3).

2.5 Character inheritance

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

```
⟨/m-t⟩
⟨*m-t|ebg|zpeu|mvs⟩
```

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

⟨/m-t|ebg|zpeu|mvs⟩
⟨*m-t⟩
```

2.5.1 OT1

Glyphs that should possibly inherit settings on one side only: 012 (‘fi’ ligature), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

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

```

2.5.2 T1

Candidates here: 028 (‘fi’), 029 (‘fl’), 030 (‘ffi’), 031 (‘ffl’), 156 (‘IJ’ ligature, since LaTeX 2005/12/01 accessible as \IJ), 188 (‘ij’, \ij), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
   { encoding = T1 }
   { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
     a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
     C = {\'C,\c C,\v C},
     c = {\'c,\c c,\v c},
     D = {\v D,\DH},
     d = {\v d,\dj},
     E = {\`E,\'E,\^E,\"E,\k E,\v E},
     e = {\`e,\'e,\^e,\"e,\k e,\v e},
     f = {027}, % ff
     G = {\u G},
     g = {\u g},
     I = {\`I,\'I,\^I,\"I,\.I},
     i = {\`i,\'i,\^i,\"i,\i},
     j = {\j},
     L = {\L,\'L,\v L},
     l = {\l,\'l,\v l},
     N = {\'N,\~N,\v N},
     n = {\'n,\~n,\v n},
     O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
     o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
     R = {\'R,\v R},
     r = {\'r,\v r},
     S = {\'S,\c S,\v S,\SS},
     s = {\'s,\c s,\v s},
     T = {\c T,\v T},
     t = {\c t,\v t},
     U = {\`U,\'U,\^U,\"U,\H U,\r U},
     u = {\`u,\'u,\^u,\"u,\H u,\r u},
     Y = {\'Y,\"Y},
     y = {\'y,\"y},
     Z = {\'Z,\.Z,\v Z},
     z = {\'z,\.z,\v z}
```

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

```
%    - = {127},
  }

```

2.5.3 LY1

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

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

```

2.5.4 OT4

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

```
\DeclareCharacterInheritance
  { encoding = OT4 }
  { A = {\k A},
    a = {\k a},
    C = {\'C},
    c = {\'c},
    E = {\k E},
    e = {\k e},
    f = {011}, % ff
    i = {\i},
    j = {\j},
    L = {\L},
    l = {\l},
    N = {\'N},
    n = {\'n},
    O = {\O,\'O},
    o = {\o,\'o},
    S = {\'S},
    s = {\'s},
    Z = {\'Z,\.Z},
    z = {\'z,\.z},
    \textquotedblleft = "FF
  }

```

2.5.5 QX

The Central European QX encoding.[10] Ligatures: 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```
\DeclareCharacterInheritance
   { encoding = QX }
   { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
     a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
     C = {\'C,\c C},
     c = {\'c,\c c},
     D = {\DH},
     E = {\`E,\'E,\^E,\"E,\k E},
     e = {\`e,\'e,\^e,\"e,\k e},
     f = {011}, % ff
     I = {\`I,\'I,\^I,\"I,\k I},
     i = {\`i,\'i,\^i,\"i,\k i,\i},
     j = {\j},
     L = {\L},
     l = {\l},
     N = {\'N,\~N},
     n = {\'n,\~n},
     O = {\O,\`O,\'O,\^O,\~O,\"O},
     o = {\o,\`o,\'o,\^o,\~o,\"o},
```

The Romanian \textcommabelow accents are actually replacements for the \c variants, which had previously (and erroneously[11]) been included in QX encoding. They are still kept for backwards compatibility.

```
     S = {\'S,\c S,\textcommabelow S,\v S},
     s = {\'s,\c s,\textcommabelow s,\v s},
     T = {\c T,\textcommabelow T},
     t = {\c t,\textcommabelow t},
     U = {\`U,\'U,\^U,\"U,\k U},
     u = {\`u,\'u,\^u,\"u,\k u},
     Y = {\'Y,\"Y},
     y = {\'y,\"y},
     Z = {\'Z,\.Z,\v Z},
     z = {\'z,\.z,\v z},
     . = \textellipsis
   }

```

2.5.6 T5

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

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

[10] Contributed by *Maciej Eder*.

[11] Cf. https://tug.org/pipermail/tex-live/2008-August/017204.html

```
     I = {\`I,\'I,\~I,\h I,\d I},
     i = {\`i,\'i,\~i,\h i,\d i,\i},
     O = {\`O,\'O,\~O,\h O,\d O,\^O,\horn O,
          \`\Ocircumflex,\'\Ocircumflex,\~\Ocircumflex,\h\Ocircumflex,\d\Ocircumflex,
          \`\Ohorn,\'\Ohorn,\~\Ohorn,\h\Ohorn,\d\Ohorn},
     o = {\`o,\'o,\~o,\h o,\d o,\^o,\horn o,
          \`\ocircumflex,\'\ocircumflex,\~\ocircumflex,\h\ocircumflex,\d\ocircumflex,
          \`\ohorn,\'\ohorn,\~\ohorn,\h\ohorn,\d\ohorn},
     U = {\`U,\'U,\~U,\h U,\d U,\horn U,
          \`\Uhorn,\'\Uhorn,\~\Uhorn,\h\Uhorn,\d\Uhorn},
     u = {\`u,\'u,\~u,\h u,\d u,\horn u,
          \`\uhorn,\'\uhorn,\~\uhorn,\h\uhorn,\d\uhorn},
     Y = {\`Y,\'Y,\~Y,\h Y,\d Y},
     y = {\`y,\'y,\~y,\h y,\d y}
   }

```

2.5.7 EU1, EU2, TU

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

```
\DeclareCharacterInheritance
   { encoding = {TU,EU1,EU2} }
   { A = {\`A,\'A,\^A,\~A,\"A,\r A,\k A,\u A},
     a = {\`a,\'a,\^a,\~a,\"a,\r a,\k a,\u a},
     C = {\'C,\c C,\v C},
     c = {\'c,\c c,\v c},
     D = {\v D,\DH},
     d = {\v d,\dj},
     E = {\`E,\'E,\^E,\"E,\k E,\v E},
     e = {\`e,\'e,\^e,\"e,\k e,\v e},
%      f = {/f_f}, % sometimes /f_f, sometimes /ff
     G = {\u G},
     g = {\u g},
     I = {\`I,\'I,\^I,\"I,\.I},
     i = {\`i,\'i,\^i,\"i,\i},
%      j = {\j},
     L = {\L,\'L,\v L},
     l = {\l,\'l,\v l},
     N = {\'N,\~N,\v N},
     n = {\'n,\~n,\v n},
     O = {\O,\`O,\'O,\^O,\~O,\"O,\H O},
     o = {\o,\`o,\'o,\^o,\~o,\"o,\H o},
     R = {\'R,\v R},
     r = {\'r,\v r},
     S = {\'S,\c S,\v S}, % \SS
     s = {\'s,\c s,\v s},
     T = {\c T,\v T},
     t = {\c t,\v t},
     U = {\`U,\'U,\^U,\"U,\H U,\r U},
     u = {\`u,\'u,\^u,\"u,\H u,\r u},
     Y = {\'Y,\"Y},
     y = {\'y,\"y},
     Z = {\'Z,\.Z,\v Z},
     z = {\'z,\.z,\v z}
   }

⟨/m-t⟩
```

2.5.8 LGR

The Greek LGR encoding. EB Garamond contains some more glyphs.

```
⟨*m-t|ebg⟩
\DeclareCharacterInheritance
  { encoding = LGR,
⟨ebg⟩    family = {EBGaramond-OsF,EBGaramond-TOsF,EBGaramond-LF,EBGaramond-TLF}
  }
  {
⟨m-t⟩    A = {012},
⟨ebg⟩    A = {009,012,253},
⟨ebg⟩ (1)E = {199},
⟨ebg⟩    H = {010},
⟨ebg⟩ (1)H = {159},
    I = {219},
⟨ebg⟩ (1)I = {155},
    O = J,
⟨ebg⟩ (1)O = {151},
    U = {013,223},
    W = {011},
    a = {014,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,
         144,145,146,148,149,150,248},
    e = {224,225,226,227,232,233,234,235},
    h = {152,153,154,156,157,158,160,161,162,163,164,165,166,167,168,169,170,
         171,172,173,174,175,249},
⟨m-t⟩    i = {200,201,202,203,208,209,210,211,216,217,218,240,241,242,243},
⟨ebg⟩    i = {008,200,201,202,203,208,209,210,211,216,217,218,240,241,242,243},
    o = {228,229,230,231,236,237,238,239},
    r = {251,252},
    u = {015,204,205,206,207,212,213,214,215,220,221,222,244,245,246,247},
    w = {176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,
         193,194,196,197,198,250},
⟨ebg⟩    \textstigma = \textvarstigma,
    . = {059}  % ano teleia
  }

⟨/m-t|ebg⟩
```

2.5.9 Euro symbols

Make Euro symbols settings simpler.

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

⟨/zpeu⟩
⟨*mvs⟩
```

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

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

⟨/mvs⟩
```

2.6 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. With XeTeX, we reset all ligatures (keeping only the TeX pseudo-ligatures).

```
⟨*m-t⟩
%%% ----------------------------------------------------------------------
%%% TRACKING/LETTERSPACING

\ifx\XeTeXrevision\@undefined
\SetTracking % pdftex/luatex
  [ name          = default,
    no ligatures = {f} ]
  { encoding      = {OT1,T1,T2A,LY1,OT4,QX,EU2,TU} }
  { }
\else
\SetTracking % xetex
  [ name          = default,
    features      = {ResetAll} ]
  { encoding      = {EU1,TU} }
  { }
\fi

```

2.7 Font expansion

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

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

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

Settings for Cyrillic T2A encoding.[12]

```
\SetExpansion
   [ name     = T2A ]
   { encoding = T2A }
   {
     A = 500,     a = 700,
     B = 700,     b = 700,
     C = 700,     c = 700,
     D = 500,     d = 700,
     E = 700,     e = 700,
     F = 700,
     G = 500,     g = 700,
     H = 700,     h = 700,
     K = 700,     k = 700,
     M = 700,     m = 700,
     N = 700,     n = 700,
     O = 500,     o = 700,
     P = 700,     p = 700,
     Q = 500,     q = 700,
     R = 700,
     S = 700,     s = 700,
     U = 700,     u = 700,
     W = 700,     w = 700,
     Z = 700,     z = 700,
     2 = 700,
     3 = 700,
     6 = 700,
     8 = 700,
     9 = 700,
     \CYRA = 500,     \cyra = 700,
     \CYRB = 700,     \cyrb = 700,
     \CYRV = 700,     \cyrv = 700,
     \CYRG = 700,     \cyrg = 700,
     \CYRD = 700,     \cyrd = 700,
     \CYRE = 700,     \cyre = 700,
     \CYRZH = 700,    \cyrzh = 700,
     \CYRZ = 700,     \cyrz = 700,
     \CYRI = 700,     \cyri = 700,
     \CYRISHRT = 700, \cyrishrt = 700,
     \CYRK = 700,     \cyrk = 700,
     \CYRL = 700,     \cyrl = 700,
     \CYRM = 700,     \cyrm = 700,
     \CYRN = 700,     \cyrn = 700,
     \CYRO = 500,     \cyro = 700,
     \CYRP = 700,     \cyrp = 700,
     \CYRR = 700,     \cyrr = 700,
     \CYRS = 700,     \cyrs = 700,
     \CYRT = 700,     \cyrt = 700,
     \CYRU = 700,     \cyru = 700,
     \CYRF = 700,     \cyrf = 700,
     \CYRH = 700,     \cyrh = 700,
     \CYRC = 700,     \cyrc = 700,
     \CYRCH = 700,    \cyrch = 700,
     \CYRSH = 700,    \cyrsh = 700,
     \CYRSHCH = 700,  \cyrshch = 700,
     \CYRHRDSN = 700, \cyrhrdsn = 700,
     \CYRERY = 700,   \cyrery = 700,
     \CYRSFTSN = 700, \cyrsftsn = 700,
     \CYREREV = 700,  \cyrerev = 700,
     \CYRYU = 700,    \cyryu = 700,
     \CYRYA = 700,    \cyrya = 700
   }
```

[12] Contributed by *Karl Karlsson*.

T5 encoding does not contain \AE, \ae, \OE and \oe.

```
\SetExpansion
   [ name     = T5 ]
   { encoding = T5 }
   {
     A = 500,     a = 700,
     B = 700,     b = 700,
     C = 700,     c = 700,
     D = 500,     d = 700,
     E = 700,     e = 700,
     F = 700,
     G = 500,     g = 700,
     H = 700,     h = 700,
     K = 700,     k = 700,
     M = 700,     m = 700,
     N = 700,     n = 700,
     O = 500,     o = 700,
     P = 700,     p = 700,
     Q = 500,     q = 700,
     R = 700,
     S = 700,     s = 700,
     U = 700,     u = 700,
     W = 700,     w = 700,
     Z = 700,     z = 700,
     2 = 700,
     3 = 700,
     6 = 700,
     8 = 700,
     9 = 700
   }

⟨/m-t⟩
```

2.8 Character protrusion

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

```

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

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

2.8.1 Normal

The default settings always use the most moderate value.

```
⟨*cfg-t⟩
\SetProtrusion
⟨m-t⟩    [ name     = default ]
```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

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

- Bitstream Letter Gothic (blg)

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

- Computer Modern Roman (cmr)

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

- EB Garamond

```
⟨ebg⟩    [ name     = EBGaramond-default ]
```

- Minion[13] (pmnx, pmnj)

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

- Palatino (ppl, pplx, pplj)

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

- Times (ptm, ptmx, ptmj)

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

- URW Garamond (ugm)

```
⟨ugm⟩    [ name     = ugm-default ]
⟨m-t|cmr|pmn|ebg⟩    { }
⟨bch|blg|ugm⟩    { encoding = OT1,
⟨ppl|ptm⟩    { encoding = {OT1,OT4},
⟨bch⟩      family   = bch }
⟨blg⟩      family   = blg }
⟨ppl⟩      family   = {ppl,pplx,pplj} }
⟨ptm⟩      family   = {ptm,ptmx,ptmj} }
⟨ugm⟩      family   = ugm }
   {
⟨m-t|bch|blg|cmr|ebg|pmn|ppl|ptm⟩      A = {50,50},
⟨ugm⟩      A = {50,100},
⟨ebg|ptm⟩    \AE = {50,  },
⟨ugm⟩    \AE = {150,50},
⟨ugm⟩      B = {  ,50},
⟨bch|ebg|pmn|ugm⟩      C = {50,  },
```

[13] Contributed by *Harald Harders* and *Karl Karlsson*.

```
⟨bch|ebg|pmn⟩      D = {  ,50},
⟨ugm⟩      D = {  ,70},
⟨ugm⟩      E = {  ,50},
⟨m-t|bch|cmr|ebg|pmn|ptm⟩      F = {  ,50},
⟨ugm⟩      F = {  ,70},
⟨bch|ebg|pmn⟩      G = {50,  },
⟨ugm⟩      G = {50,50},
⟨blg⟩      I = {150,150},
⟨m-t|cmr|ebg|pmn|ppl|ptm|ugm⟩      J = {50,  },
⟨bch|blg⟩      J = {100,  },
⟨!blg⟩      K = {  ,50},
⟨blg⟩      K = {50,  },
⟨m-t|bch|cmr|ebg|pmn|ppl⟩      L = {  ,50},
⟨blg⟩      L = {  ,150},
⟨ptm⟩      L = {  ,80},
⟨ugm⟩      L = {  ,120},
⟨bch|ebg|pmn|ugm⟩      O = {50,50},
⟨ebg⟩    \OE = {50,  },
⟨ugm⟩    \OE = {50,50},
⟨blg⟩      P = {  ,100},
⟨ugm⟩      P = {  ,50},
⟨bch|ebg|pmn⟩      Q = {50,70},
⟨ugm⟩      Q = {50,50},
⟨bch⟩      R = {  ,50},
⟨ugm|ebg⟩      R = {  ,70},
⟨m-t|bch|cmr|pmn|ppl|ptm⟩      T = {50,50},
⟨blg⟩      T = {100,100},
⟨ebg|ugm⟩      T = {70,70},
⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩      V = {50,50},
⟨blg|ugm⟩      V = {70,70},
⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩      W = {50,50},
⟨ugm⟩      W = {70,70},
⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩      X = {50,50},
⟨ugm⟩      X = {50,70},
⟨m-t|bch|cmr|ebg|pmn|ppl⟩      Y = {50,50},
⟨blg|ptm|ugm⟩      Y = {80,80},
⟨ugm⟩      Z = {50,50},
⟨blg⟩      f = {150,100},
⟨blg⟩      i = {150,150},
⟨blg⟩      j = {100,100},
⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩      k = {  ,50},
⟨ugm⟩      k = {  ,70},
⟨blg⟩      l = {150,150},
⟨pmn⟩      l = {  ,-50},
⟨ppl⟩      p = {50,50},
⟨ebg|ugm⟩      p = {  ,50},
⟨ebg|ppl⟩      q = {50,  },
⟨!blg⟩      r = {  ,50},
⟨blg⟩      r = {100, 80},
⟨cmr|ebg|pmn⟩      t = {  ,70},
⟨bch⟩      t = {  ,50},
⟨blg⟩      t = {150, 80},
⟨ugm⟩      t = {  ,100},
⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩      v = {50,50},
⟨blg⟩      v = {100,100},
⟨ugm⟩      v = {50,70},
⟨m-t|bch|cmr|ebg|pmn|ppl|ptm⟩      w = {50,50},
⟨ugm⟩      w = {50,70},
⟨!blg⟩      x = {50,50},
⟨blg⟩      x = {100,100},
⟨m-t|bch|ebg|pmn⟩      y = {  ,50},
⟨blg⟩      y = { 50,100},
⟨cmr|ppl|ptm⟩      y = {50,70},
⟨ugm⟩      y = {  ,70},

⟨cmr⟩      0 = {  ,50},
⟨m-t⟩      1 = {50,50},
⟨bch|blg|ptm|ugm⟩      1 = {150,150},
⟨cmr⟩      1 = {100,200},
⟨pmn⟩      1 = {  ,50},
⟨ppl⟩      1 = {100,100},
⟨bch|cmr|ugm⟩      2 = {50,50},
⟨blg⟩      2 = {  ,100},
⟨bch|pmn⟩      3 = {50,  },
⟨cmr|ugm⟩      3 = {50,50},
⟨blg⟩      3 = {100,  },
⟨m-t⟩      4 = {50,50},
⟨bch⟩      4 = {100,50},
⟨blg⟩      4 = {100,  },
⟨cmr|ugm⟩      4 = {70,70},
⟨pmn⟩      4 = {50,  },
⟨ptm⟩      4 = {70,  },
⟨cmr⟩      5 = {  ,50},
⟨bch⟩      6 = {50,  },
⟨cmr⟩      6 = {  ,50},
⟨m-t⟩      7 = {50,50},
⟨bch|pmn|ugm⟩      7 = {50,80},
⟨blg⟩      7 = {100,100},
⟨cmr|ptm⟩      7 = {50,100},
⟨ppl⟩      7 = {  ,50},
⟨cmr⟩      8 = {  ,50},
⟨bch⟩      9 = {50,50},
⟨cmr⟩      9 = {  ,50},
⟨m-t|cmr|pmn|ppl|ptm|ugm⟩      . = { ,700},
⟨bch|ebg⟩      . = { ,600},
⟨blg⟩      . = {400,500},
⟨!blg⟩     {,}= { ,500},
⟨blg⟩     {,}= {300,400},
⟨m-t|cmr|pmn|ppl|ptm|ugm⟩      : = { ,500},
⟨bch|ebg⟩      : = { ,400},
⟨blg⟩      : = {300,400},
⟨m-t|bch|ebg|pmn|ptm⟩      ; = { ,300},
⟨blg⟩      ; = {200,300},
⟨cmr|ppl⟩      ; = { ,500},
⟨ugm⟩      ; = { ,400},
⟨!blg⟩      ! = { ,100},
⟨blg⟩      ! = {200,200},
⟨m-t|ebg|pmn|ptm⟩      ? = { ,100},
⟨bch|cmr|ppl|ugm⟩      ? = { ,200},
⟨blg⟩      ? = {150,150},
⟨pmn⟩      " = {300,300},
⟨m-t|bch|cmr|ebg|pmn|ppl⟩      @ = {50,50},
⟨ptm⟩      @ = {100,100},
⟨m-t|bch|blg|cmr|ebg|pmn|ppl|ptm⟩      ~ = {200,250},
⟨ugm⟩      ~ = {300,350},
⟨ebg|ppl|ptm⟩      & = {50,100},
⟨ugm⟩      & = {  ,100},
⟨m-t|cmr|ebg|pmn⟩     \% = {50,50},
⟨bch⟩     \% = {  ,50},
⟨ppl|ptm⟩     \% = {100,100},
⟨ugm⟩     \% = {50,100},
⟨blg⟩     \# = {100,100},
⟨m-t|ppl|ptm|ugm⟩      * = {200,200},
⟨bch|pmn⟩      * = {200,300},
⟨blg⟩      * = {150,200},
⟨cmr|ebg⟩      * = {300,300},
⟨m-t|cmr|ebg|ppl|ptm⟩      + = {250,250},
⟨bch⟩      + = {150,250},
⟨blg|pmn⟩      + = {150,200},
⟨ugm⟩      + = {250,300},
⟨blg|ugm⟩     {=}= {200,200},
⟨m-t|ebg|pmn|ptm⟩     ( = {100,   },    ) = {   ,200},
⟨bch|ugm⟩     ( = {200,   },    ) = {   ,200},
⟨cmr|blg⟩     ( = {300,   },    ) = {   ,300},
⟨ppl⟩     ( = {100,   },    ) = {   ,300},
⟨bch|pmn⟩     [ = {100,   },    ] = {   ,100},
⟨blg⟩     [ = {300,100},    ] = {   ,300},

⟨m-t|ebg|pmn|ptm⟩     / = {100,200},
⟨bch⟩     / = { ,200},
⟨blg⟩     / = {300,300},
⟨cmr|ppl⟩     / = {200,300},
⟨ugm⟩     / = {100,300},
⟨m-t|ptm⟩     - = {500,500},
⟨bch|cmr|ppl⟩     - = {400,500},
⟨blg⟩     - = {300,400},
⟨ebg⟩     - = {300,500},
⟨pmn⟩     - = {200,400},
⟨ugm⟩     - = {500,600},
⟨blg⟩     < = {200,100},    > = {100,200},
⟨blg⟩     _ = {150,250},
⟨blg⟩     | = {250,250},
⟨m-t|pmn⟩     \textendash       = {200,200},   \textemdash        = {150,150},
⟨bch⟩     \textendash       = {200,300},   \textemdash        = {150,250},
⟨cmr⟩     \textendash       = {400,300},   \textemdash        = {300,200},
⟨ebg|ppl|ptm⟩     \textendash       = {300,300},   \textemdash        = {200,200},
⟨ugm⟩     \textendash       = {250,300},   \textemdash        = {250,250},
```

Why settings for left *and* right quotes? Because in some languages they might be used like that (see the csquotes package for examples).

```
⟨m-t|bch|pmn⟩     \textquoteleft    = {300,400},   \textquoteright    = {300,400},
⟨blg⟩     \textquoteleft    = {400,600},   \textquoteright    = {400,600},
⟨cmr⟩     \textquoteleft    = {500,700},   \textquoteright    = {500,600},
⟨ebg⟩     \textquoteleft    = {300,500},   \textquoteright    = {400,400},
⟨ppl⟩     \textquoteleft    = {500,700},   \textquoteright    = {500,700},
⟨ptm⟩     \textquoteleft    = {500,500},   \textquoteright    = {300,500},
⟨ugm⟩     \textquoteleft    = {300,600},   \textquoteright    = {300,600},
⟨m-t|ebg|bch|pmn⟩     \textquotedblleft = {300,300},   \textquotedblright = {300,300}
⟨blg⟩     \textquotedblright = {300,400}
⟨cmr⟩     \textquotedblleft = {500,300},   \textquotedblright = {200,600}
⟨ppl|ptm⟩     \textquotedblleft = {300,400},   \textquotedblright = {300,400}
⟨ugm⟩     \textquotedblleft = {400,400},   \textquotedblright = {400,400}
   }

```

Greek uppercase letters are in OT1 encoding only.

```
⟨*m-t|cmr|ebg|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = OT1-default,
⟨cmr⟩   [ name     = cmr-OT1,
⟨ebg⟩   [ name     = EBGaramond-OT1,
⟨pmn⟩   [ name     = pmnj-OT1,
⟨m-t⟩     load     = default ]
⟨cmr⟩     load     = cmr-default ]
⟨ebg⟩     load     = EBGaramond-default ]
⟨pmn⟩     load     = pmnj-default ]
⟨m-t⟩   { encoding = OT1 }
⟨cmr⟩   { encoding = {OT1,OT4},
⟨pmn⟩   { encoding = OT1,
⟨cmr⟩     family   = cmr  }
⟨pmn⟩     family   = pmnj }
⟨ebg⟩   { }
   {
⟨m-t|cmr⟩     \AE = {50,   },
⟨pmn⟩     \OE = {50,  }
⟨*cmr|ebg⟩
```

```
     "00 = {    ,150}, % \Gamma
     "01 = {100,100}, % \Delta
     "02 = { 50, 50}, % \Theta
     "03 = {100,100}, % \Lambda
⟨ebg⟩     "04 = { 50, 50}, % \Xi
⟨cmr⟩     "06 = { 50, 50}, % \Sigma
     "07 = {100,100}, % \Upsilon
     "08 = { 50, 50}, % \Phi
     "09 = { 50, 50}, % \Psi
⟨ebg⟩     "0A = { 50, 50}, % \Omega
⟨ebg⟩     138 = {    , 50}, % \L
```

Remaining slots can be found in the source file.

```
⟨/cmr|ebg⟩
   }

```

Settings for figure variants.

```
⟨*ebg⟩
\SetProtrusion
   [ name     = EBGaramond-OT1-LF,
     load     = EBGaramond-OT1 ]
   { encoding = OT1,
     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
   {
     1 = {50,50},
     2 = {50,50},
     4 = {50,50},
     7 = {50,50},
   }

\SetProtrusion
   [ name     = EBGaramond-OT1-TOsF,
     load     = EBGaramond-OT1 ]
   { encoding = OT1,
     family   = {EBGaramond-TOsF} }
   {
     1 = {150,150},
     2 = {50,50},
     3 = {50,50},
     4 = {50,50},
     5 = {50,50},
     6 = {50,50},
     7 = {50,80},
     8 = {50,50},
     9 = {50,50},
   }

⟨/ebg⟩
⟨/m-t|cmr|ebg|pmn⟩
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first. For XƎTEX (EU1) and LuaTEX (EU2) we simply use the T1 list as default (for now).

```
\SetProtrusion
⟨m-t⟩   [ name     = T1-default,
⟨bch⟩   [ name     = bch-T1,
⟨blg⟩   [ name     = blg-T1,
⟨cmr⟩   [ name     = cmr-T1,
⟨ebg⟩   [ name     = EBGaramond-T1,
⟨pmn⟩   [ name     = pmnj-T1,
⟨ppl⟩   [ name     = ppl-T1,
⟨ptm⟩   [ name     = ptm-T1,
⟨ugm⟩   [ name     = ugm-T1,
⟨m-t⟩     load     = default     ]
⟨bch⟩     load     = bch-default ]
⟨blg⟩     load     = blg-default ]
⟨cmr⟩     load     = cmr-default ]
⟨ebg⟩     load     = EBGaramond-default ]
⟨pmn⟩     load     = pmnj-default ]
⟨ppl⟩     load     = ppl-default ]
⟨ptm⟩     load     = ptm-default ]
⟨ugm⟩     load     = ugm-default ]
⟨m-t⟩   { encoding = {T1,LY1,EU1,EU2,TU} }
⟨bch|cmr|pmn|ppl⟩   { encoding = {T1,LY1},
⟨blg|ptm|ugm⟩   { encoding = {T1},
⟨ebg⟩   { encoding = {LY1},
⟨bch⟩     family   = bch }
⟨blg⟩     family   = blg }
⟨cmr⟩     family   = cmr }
⟨ebg⟩     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF} }
⟨pmn⟩     family   = pmnj }
⟨ppl⟩     family   = {ppl,pplx,pplj} }
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
⟨ugm⟩     family   = ugm }
    {
⟨m-t|cmr⟩     \AE = {50,  },
⟨bch|pmn⟩     \OE = {50,  },
⟨pmn⟩     \TH = {  ,50},
⟨blg⟩     \v L = {   ,250},
⟨blg⟩     \v d = {   ,250},
⟨blg⟩     \v l = {   ,250},
⟨blg⟩     \v t = {   ,250},
⟨blg⟩     127 =  {300,400},
⟨blg⟩     156 =  {100,   }, % IJ
⟨blg⟩     188 =  { 80, 80}, % ij
⟨m-t|bch|ebg|pmn|ppl|ptm⟩       _ = {100,100},
⟨cmr⟩       _ = {200,200},
⟨ugm⟩       _ = {100,200},
⟨m-t|ebg|pmn|ptm⟩     \textbackslash    = {100,200},
⟨bch⟩     \textbackslash    = {150,200},
⟨blg⟩     \textbackslash    = {250,300},
⟨cmr|ppl⟩     \textbackslash    = {200,300},
⟨ugm⟩     \textbackslash    = {100,300},
⟨ugm⟩     \textbar          = {200,200},
⟨blg⟩     \textendash       = {300,300},   \textemdash        = {150,150},
⟨blg⟩     \textquotedbl     = {300,400},   \textquotedblleft  = {300,400},
⟨cmr⟩     \textquotedbl     = {300,300},   \textquotedblleft  = {200,600},
```

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

```
⟨m-t|cmr|ebg|ppl|ptm|ugm⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨blg⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,400},
⟨bch|pmn⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨m-t|bch|pmn⟩     \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨blg⟩     \guilsinglleft    = {300,500},   \guilsinglright    = {300,500},
⟨cmr|ebg|ppl|ptm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨ugm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch|pmn⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨blg|ppl|ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ebg⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,300},
⟨ugm⟩     \guillemotleft    = {300,400},   \guillemotright    = {300,400},
⟨m-t|bch|cmr|ebg|pmn|ppl|ugm⟩     \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨blg⟩     \textexclamdown   = {200,   },   \textquestiondown  = {100,   },
⟨ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨m-t|cmr|ebg|ppl|ptm|ugm⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨bch|blg|pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t|bch|cmr|ebg|ppl|ptm|ugm⟩     \textless         = {200,100},   \textgreater       = {100,200}
```

```
⟨pmn⟩    \textless         = {100,   },   \textgreater       = {   ,100},
⟨pmn⟩    \textvisiblespace = {100,100} % not in LY1
   }

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```
⟨*cmr⟩
\SetProtrusion
   [ name     = lmr-T1,
     load     = cmr-T1   ]
   { encoding = {T1,LY1},
     family   = lmr      }
   {
     \textquotedblleft  = {300,400},  \textquotedblright = {300,400}
   }

⟨/cmr⟩
⟨*ebg⟩
\SetProtrusion
   [ name     = EBGaramond-T1-LF,
     load     = EBGaramond-T1 ]
   { encoding = T1,
     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
   {
     1 = {50,50},
     2 = {50,50},
     4 = {50,50},
     7 = {50,50},
   }

\SetProtrusion
   [ name     = EBGaramond-T1-TOsF,
     load     = EBGaramond-T1 ]
   { encoding = T1,
     family   = {EBGaramond-TOsF} }
   {
     1 = {150,150},
     2 = {50,50},
     3 = {50,50},
     4 = {50,50},
     5 = {50,50},
     6 = {50,50},
     7 = {50,80},
     8 = {50,50},
     9 = {50,50},
   }

⟨/ebg⟩
```

Settings for the T2A encoding (generic, Computer Modern Roman, and Minion).[14]

```
⟨*m-t|cmr|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = T2A-default,
⟨cmr⟩   [ name     = cmr-T2A,
⟨pmn⟩   [ name     = pmnj-T2A,
⟨m-t⟩     load     = default     ]
⟨cmr⟩     load     = cmr-default ]
⟨pmn⟩     load     = pmnj-default ]
   { encoding = T2A,
⟨m-t⟩   }
⟨cmr⟩     family   = cmr }
⟨pmn⟩     family   = pmnj }
```

[14] Contributed by *Karl Karlsson*.

```
   {
     \CYRA = {50,50},
     \CYRG = {  ,50},
     \CYRK = {  ,50},
     \CYRT = {50,50},
     \CYRH = {50,50},
     \CYRU = {50,50},
⟨pmn⟩     \CYRS = {50,  },
⟨pmn⟩     \CYRO = {50,50},
     \cyrk = {  ,50},
     \cyrg = {  ,50},
     \cyrh = {50,50},
⟨m-t|pmn⟩     \cyru = {50,50},
⟨cmr⟩     \cyru = {50,70},
⟨m-t⟩        _  = {100,100},
⟨cmr⟩        _  = {200,200},
⟨m-t⟩     \textbackslash    = {100,200},   \quotedblbase      = {400,400},
⟨cmr⟩     \textbackslash    = {200,300},   \quotedblbase      = {400,400},
⟨pmn⟩     \textbackslash    = {100,200},   \quotedblbase      = {300,300},
⟨cmr⟩     \textquotedbl     = {300,300},   \textquotedblleft  = {200,600},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨pmn⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨m-t|cmr⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨m-t|cmr⟩     \textless         = {200,100},   \textgreater       = {100,200}
⟨pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100}
   }

⟨/m-t|cmr|pmn⟩
```

Settings for the QX encoding (generic and Times).[15] It also includes some glyphs otherwise in TS1.

```
⟨*m-t|ptm⟩
\SetProtrusion
⟨m-t⟩   [ name     = QX-default,
⟨ptm⟩   [ name     = ptm-QX,
⟨m-t⟩     load     = default ]
⟨ptm⟩     load     = ptm-default ]
⟨m-t⟩   { encoding = QX }
⟨ptm⟩   { encoding = QX,
⟨ptm⟩     family   = {ptm,ptmx,ptmj} }
   {
     \AE = {50,  },
⟨ptm⟩      *  = {200,200},
     {=} = {100,100},
     \textunderscore   = {100,100},
     \textbackslash    = {100,200},
     \quotedblbase     = {400,400},
⟨m-t⟩     \guillemotleft    = {200,200},   \guillemotright    = {200,200},
⟨ptm⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
     \textexclamdown   = {100,   },   \textquestiondown  = {100,   },
⟨m-t⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
⟨ptm⟩     \textbraceleft    = {200,200},   \textbraceright    = {200,300},
     \textless         = {200,100},   \textgreater       = {100,200},
     \textminus        = {200,200},   \textdegree        = {300,300},
⟨m-t⟩     \copyright        = {100,100},   \textregistered    = {100,100}
⟨ptm⟩     \copyright        = {100,150},   \textregistered    = {100,150},
⟨ptm⟩     \textxgeq         = {   ,100},   \textxleq          = {100,   },
⟨ptm⟩     \textalpha        = {   , 50},   \textDelta         = { 70, 70},
⟨ptm⟩     \textpi           = { 50, 80},   \textSigma         = {   , 70},
⟨ptm⟩     \textmu           = {   , 80},   \texteuro          = { 50, 50},
⟨ptm⟩     \textellipsis     = {150,200},   \textasciitilde    = { 80, 80},
```

[15] Contributed by *Maciej Eder*.

```
⟨ptm⟩     \textapprox       = { 50, 50},   \textinfty         = {100,100},
⟨ptm⟩     \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
⟨ptm⟩     \textdiv          = { 50,150},   \textsection       = { 80, 80},
⟨ptm⟩     \texttimes        = {100,150},   \textpm            = { 50, 80},
⟨ptm⟩     \textbullet       = {150,150},   \textperiodcentered = {300,300},
⟨ptm⟩     \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
⟨ptm⟩     \textperthousand  = {    ,50}
   }

⟨/m-t|ptm⟩
```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented characters are already taken care of by the inheritance list.

```
⟨*cmr|bch⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-T5,
⟨cmr⟩     load     = cmr-default ]
⟨bch⟩   [ name     = bch-T5,
⟨bch⟩     load     = bch-default ]
   { encoding = T5,
⟨cmr⟩     family   = cmr }
⟨bch⟩     family   = bch }
   {
⟨bch⟩     _ = {100,100},
⟨bch⟩     \textbackslash    = {150,200},
⟨cmr⟩     \textbackslash    = {200,300},
⟨cmr⟩     \textquotedblleft = {200,600},
⟨cmr⟩     \textquotedbl     = {300,300},
⟨bch⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {300,300},
⟨cmr⟩     \quotesinglbase   = {400,400},   \quotedblbase      = {400,400},
⟨bch⟩     \guilsinglleft    = {400,300},   \guilsinglright    = {300,400},
⟨cmr⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨bch⟩     \guillemotleft    = {200,200},   \guillemotright    = {150,300},
⟨cmr⟩     \guillemotleft    = {300,200},   \guillemotright    = {100,400},
⟨bch⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,300},
⟨cmr⟩     \textbraceleft    = {400,200},   \textbraceright    = {200,400},
     \textless         = {200,100},   \textgreater       = {100,200}
   }

⟨/cmr|bch⟩
```

Minion with lining numbers.

```
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-OT1,
     load     = pmnj-default ]
   { encoding = OT1,
     family   = pmnx }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-T1,
     load     = pmnj-T1  ]
   { encoding = {T1,LY1},
     family   = pmnx     }
   {
     1 = {230,180}
   }

\SetProtrusion
   [ name     = pmnx-T2A,
     load     = pmnj-T2A  ]
   { encoding = {T2A},
     family   = pmnx     }
   {
     1 = {230,180}
   }

⟨/pmn⟩
```

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

```
⟨*ptm⟩
 \SetProtrusion
   [ name     = ptm-LY1,
     load     = ptm-T1 ]
   { encoding = LY1,
     family   = {ptm,ptmx,ptmj} }
   {
     _                         = {100,100},
     \texttrademark            = {100,100},
     \textregistered           = {100,100},
     \textcopyright            = {100,100},
     \textdegree               = {300,300},
     \textminus                = {200,200},
     \textellipsis             = {150,200},
%    \texteuro                 = {   ,   }, % ?
     \textcent                 = {100,100},
     \textquotesingle          = {500,500},
     \textflorin               = { 50, 70},
     \textdagger               = {150,150},
     \textdaggerdbl            = {100,100},
     \textperthousand          = {   , 50},
     \textbullet               = {150,150},
     \textonesuperior          = {100,100},
     \texttwosuperior          = { 50, 50},
     \textthreesuperior        = { 50, 50},
     \textperiodcentered       = {300,300},
     \textplusminus            = { 50, 80},
     \textmultiply             = {100,100},
     \textdivide               = { 50,150}
```

Remaining slots in the source file.

```
   }

⟨/ptm⟩
```

For the Greek LGR encoding.

```
⟨*ebg⟩
\SetProtrusion
  [ name = EBGaramond-LGR ]
  { }
  {
     A = {50,50},
     D = {100,100},
     F = {50,50},
     G = {  ,150},
     K = {  ,50},
     L = {100,100},
     O = {50,50},
     U = {100,100},
     T = {50,50},
     W = {  ,50},
     Y = {50,50},
     . = { ,600},
    {,}= { ,500},
     : = { ,400},
     ; = { ,300},
     ! = { ,100},
     ? = { ,100},
     ~ = {200,250},
    \% = {50,50},
     * = {300,300},
     + = {250,250},
    {=}= { 50, 50},
     ( = {100,   },    ) = {   ,200},
     / = {100,200},
     - = {300,500},
     \texteuro = { 50,100},
     \textendash       = {300,300},   \textemdash        = {200,200},
     \textquoteleft    = {300,500},   \textquoteright    = {400,400},
     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
  }

\SetProtrusion
   [ name     = EBGaramond-LGR-LF,
     load     = EBGaramond-LGR ]
   { encoding = LGR,
     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF} }
   {
     1 = {50,50},
     2 = {50,50},
     4 = {50,50},
     7 = {50,50},
   }

\SetProtrusion
   [ name     = EBGaramond-LGR-TOsF,
     load     = EBGaramond-LGR ]
   { encoding = LGR,
     family   = {EBGaramond-TOsF} }
   {
     1 = {150,150},
     2 = {50,50},
     3 = {50,50},
     4 = {50,50},
     5 = {50,50},
     6 = {50,50},
     7 = {50,80},
     8 = {50,50},
     9 = {50,50},
   }

⟨/ebg⟩
```

2.8.2 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.[16]

```
\SetProtrusion
⟨m-t⟩  [ name     = OT1-it   ]
⟨bch⟩  [ name     = bch-it   ]
⟨blg⟩  [ name     = blg-it,
⟨blg⟩    load     = blg-default ]
```

[16] Settings contributed by *Hendrik Vogt*.

```
⟨cmr⟩  [ name     = cmr-it   ]
⟨ebg⟩  [ name     = EBGaramond-it   ]
⟨pmn⟩  [ name     = pmnj-it  ]
⟨ppl⟩  [ name     = ppl-it   ]
⟨ptm⟩  [ name     = ptm-it   ]
⟨ugm⟩  [ name     = ugm-it   ]
⟨m-t|bch|blg|ugm⟩  { encoding = OT1,
⟨ppl|ptm⟩  { encoding = {OT1,OT4},
⟨bch⟩    family   = bch,
⟨blg⟩    family   = blg,
⟨ppl⟩    family   = {ppl,pplx,pplj},
⟨ptm⟩    family   = {ptm,ptmx,ptmj},
⟨ugm⟩    family   = ugm,
⟨m-t|bch|ppl|ptm⟩    shape    = {it,sl}  }
⟨blg|ugm⟩    shape    = it  }
⟨cmr|ebg|pmn⟩  { }
  {
⟨cmr⟩    A = {100,100},
⟨ptm⟩    A = {100,50},
⟨ebg|pmn⟩    A = {50,  },
⟨ugm⟩    A = {  ,150},
⟨ppl⟩    A = {50,50},
⟨ptm⟩  \AE = {100,  },
⟨ebg|ppl⟩  \AE = {50,  },
⟨cmr⟩    B = {83,-40},
⟨ebg|ppl|ptm⟩    B = {50,  },
⟨pmn⟩    B = {20,-50},
⟨bch|ppl|ptm|ugm⟩    C = {50,  },
⟨cmr⟩    C = {165,-75},
⟨ebg⟩    C = {100, },
⟨pmn⟩    C = {50,-50},
⟨cmr⟩    D = {75, -28},
⟨ebg|ppl|ptm⟩    D = {50,50},
⟨pmn⟩    D = {20,  },
⟨cmr⟩    E = {80,-55},
⟨ebg|ppl|ptm⟩    E = {50,  },
⟨pmn⟩    E = {20,-50},
⟨cmr⟩    F = {85,-80},
⟨ebg|ptm⟩    F = {100, },
⟨pmn⟩    F = {10,  },
⟨ppl⟩    F = {50,  },
⟨bch|ppl|ptm|ugm⟩    G = {50,  },
⟨cmr⟩    G = {153,-15},
⟨ebg⟩    G = {100, },
⟨pmn⟩    G = {50,-50},
⟨cmr⟩    H = {73,-60},
⟨ebg|ppl|ptm⟩    H = {50,  },
⟨cmr⟩    I = {140,-120},
⟨ebg|ptm⟩    I = {50,  },
⟨pmn⟩    I = {20,-50},
⟨cmr⟩    J = {135,-80},
⟨ebg⟩    J = {50,  },
⟨pmn⟩    J = {20,  },
⟨ptm⟩    J = {100, },
⟨cmr⟩    K = {70,-30},
⟨ebg|ppl|ptm⟩    K = {50,  },
⟨pmn⟩    K = {20,  },
⟨cmr⟩    L = {87, 40},
⟨ebg|ppl|ptm⟩    L = {50,  },
⟨pmn⟩    L = {20,50},
⟨ugm⟩    L = {  ,100},
⟨cmr⟩    M = {67,-45},
⟨pmn⟩    M = {  ,-30},
⟨ptm⟩    M = {50,  },
⟨cmr⟩    N = {75,-55},
⟨pmn⟩      N = {  ,-30},
⟨ptm⟩      N = {50,  },
⟨bch|pmn|ppl|ptm⟩      O = {50,  },
⟨cmr⟩      O = {150,-30},
⟨ebg⟩      O = {100, },
⟨ugm⟩      O = {70,50},
⟨ppl|ptm⟩    \OE = {50,  },
⟨ebg⟩    \OE = {100, },
⟨cmr⟩      P = {82,-50},
⟨ebg|ppl|ptm⟩      P = {50,  },
⟨pmn⟩      P = {20,-50},
⟨bch|pmn|ppl|ptm⟩      Q = {50,  },
⟨cmr⟩      Q = {150,-30},
⟨ebg⟩      Q = {100, },
⟨ugm⟩      Q = {70,50},
⟨cmr⟩      R = {75, 15},
⟨ebg|ppl|ptm⟩      R = {50,  },
⟨pmn⟩      R = {20,  },
⟨bch|ebg|ppl|ptm⟩      S = {50,  },
⟨cmr⟩      S = {90,-65},
⟨pmn⟩      S = {20,-30},
⟨bch|ebg|ppl|ptm⟩      $ = {50,  },
⟨cmr⟩      $ = {100,-20},
⟨pmn⟩      $ = {20,-30},
⟨bch|pmn|ugm⟩      T = {70,  },
⟨cmr⟩      T = {220,-85},
⟨ebg|ppl|ptm⟩      T = {100, },
⟨cmr⟩      U = {230,-55},
⟨ebg|ppl|ptm⟩      U = {50,  },
⟨pmn⟩      U = {50,-50},
⟨cmr⟩      V = {260,-60},
⟨ebg|pmn|ugm⟩      V = {100, },
⟨ppl|ptm⟩      V = {100,50},
⟨cmr⟩      W = {185,-55},
⟨ebg|pmn|ugm⟩      W = {100, },
⟨ppl⟩      W = {50,  },
⟨ptm⟩      W = {100,50},
⟨cmr⟩      X = {70,-30},
⟨ppl|ptm⟩      X = {50,  },
⟨cmr⟩      Y = {250,-60},
⟨pmn⟩      Y = {50,  },
⟨ppl⟩      Y = {100,50},
⟨ptm⟩      Y = {100, },
⟨cmr⟩      Z = {90,-60},
⟨pmn⟩      Z = {  ,-50},
⟨cmr⟩      a = {150,-10},
⟨cmr⟩      b = {170,   },
⟨cmr⟩      c = {173,-10},
⟨cmr⟩      d = {150,-55},
⟨pmn⟩      d = {  ,-50},
⟨cmr⟩      e = {180, },
⟨cmr⟩      f = {  ,-250},
⟨ebg|pmn⟩      f = { ,-100},
⟨cmr⟩      g = {150,-10},
⟨cmr⟩      h = {100, },
⟨cmr⟩      i = {210, },
⟨pmn⟩      i = {  ,-30},
⟨cmr⟩      j = {  ,-40},
⟨pmn⟩      j = {  ,-30},
⟨cmr⟩      k = {110,-50},
⟨cmr⟩      l = {240,-110},
⟨pmn⟩      l = {  ,-100},
⟨cmr⟩      m = {80, },
⟨cmr⟩      n = {115, },
⟨bch⟩      o = {50,50},
⟨cmr⟩      o = {155, },
⟨bch⟩      p = {  ,50},
⟨pmn⟩      p = {-50,  },
⟨bch⟩      q = {50,  },
⟨cmr⟩      q = {170,-40},
⟨cmr⟩      r = {155,-40},
⟨pmn⟩      r = {  ,50},
⟨cmr⟩      s = {130, },
⟨bch⟩      t = {  ,50},
⟨cmr⟩      t = {230,-10},
⟨cmr⟩      u = {120, },
⟨cmr⟩      v = {140,-25},
⟨pmn|ugm⟩      v = {50,  },
⟨bch⟩      w = {  ,50},
⟨cmr⟩      w = {98,-20},
⟨pmn|ugm⟩      w = {50,  },
⟨cmr⟩      x = {65,-40},
⟨bch⟩      y = {  ,50},
⟨cmr⟩      y = {130,-20},
⟨cmr⟩      z = {110,-80},
⟨cmr⟩      0 = {170,-85},
⟨bch|ptm⟩      1 = {150,100},
⟨cmr⟩      1 = {230,110},
⟨ebg⟩      1 = {150, },
⟨pmn⟩      1 = {50,  },
⟨ppl⟩      1 = {100, },
⟨ugm⟩      1 = {150,150},
⟨cmr⟩      2 = {130,-70},
⟨ebg|ppl|ptm⟩      2 = {50,  },
⟨pmn⟩      2 = {-50,  },
⟨bch⟩      3 = {50,  },
⟨cmr⟩      3 = {140,-70},
⟨pmn⟩      3 = {-100, },
⟨ptm⟩      3 = {100,50},
⟨bch⟩      4 = {100, },
⟨cmr⟩      4 = {130,80},
⟨ebg⟩      4 = {150, },
⟨ppl|ptm⟩      4 = {50,  },
⟨cmr⟩      5 = {160, },
⟨ptm⟩      5 = {50,  },
⟨bch⟩      6 = {50,  },
⟨cmr⟩      6 = {175,-30},
⟨bch|ebg|ptm⟩      7 = {100, },
⟨cmr⟩      7 = {250,-150},
⟨pmn⟩      7 = {20,  },
⟨ppl⟩      7 = {50,  },
⟨cmr⟩      8 = {130,-40},
⟨cmr⟩      9 = {155,-80},
⟨m-t|cmr|ebg|pmn|ppl⟩      . = { ,500},
⟨blg⟩      . = {400,600},
⟨bch|ptm|ugm⟩      . = { ,700},
⟨blg⟩     {,}= {300,500},
⟨m-t|ebg|pmn|ppl⟩     {,}= { ,500},
⟨cmr⟩     {,}= { ,450},
⟨bch|ugm⟩     {,}= { ,600},
⟨ptm⟩     {,}= { ,700},
⟨m-t|cmr|ebg|ppl⟩      : = { ,300},
⟨bch|ugm⟩      : = { ,400},
⟨pmn⟩      : = { ,200},
⟨ptm⟩      : = { ,500},
⟨m-t|cmr|ebg|ppl⟩      ; = { ,300},
⟨bch|ugm⟩      ; = { ,400},
⟨pmn⟩      ; = { ,200},
⟨ptm⟩      ; = { ,500},
⟨ptm⟩      ! = { ,100},
⟨bch⟩      ? = { ,200},
⟨ptm⟩      ? = { ,100},
⟨ppl⟩      ? = { ,300},
⟨pmn⟩      " = {400,200},
⟨m-t|ebg|pmn|ppl|ptm⟩     & = {50,50},
⟨bch⟩      & = {  ,80},
⟨cmr⟩      & = {130,30},
⟨ugm⟩      & = {50,100},
⟨m-t|ebg|pmn⟩    \% = {100, },
⟨cmr⟩     \% = {180,50},
⟨bch⟩     \% = {50,50},
⟨ppl|ptm⟩     \% = {100,100},
⟨ugm⟩     \% = {100,50},
⟨m-t|pmn|ppl⟩     * = {200,200},
⟨bch⟩      * = {300,200},
⟨cmr⟩      * = {380,20},
⟨ebg⟩      * = {500,100},
⟨ptm|ugm⟩      * = {400,200},
⟨m-t|pmn|ppl⟩     + = {150,200},
⟨cmr⟩      + = {180,200},
⟨bch|ugm⟩      + = {250,250},
⟨ebg|ptm⟩      + = {250,200},
⟨m-t|ebg|pmn|ppl⟩     @ = {50,50},
⟨bch⟩      @ = {80,50},
⟨cmr⟩      @ = {180,10},
⟨ptm⟩      @ = {150,150},
⟨m-t|bch|ugm⟩     ~ = {150,150},
⟨cmr|ebg|pmn|ppl|ptm⟩     ~ = {200,150},
⟨ugm⟩     {=}= {200,200},
⟨m-t|bch|ebg|pmn|ppl|ptm|ugm⟩     ( = {200, },    ) = {  ,200},
⟨cmr⟩      ( = {300, },    ) = {  ,70},
⟨m-t|ebg|ppl|ptm|ugm⟩     / = {100,200},
⟨cmr⟩      / = {100,100},
⟨bch⟩      / = {  ,150},
⟨pmn⟩      / = {100,150},
⟨m-t⟩      - = {300,300},
⟨bch|ebg⟩      - = {300,400},
⟨pmn⟩      - = {200,300},
⟨cmr⟩      - = {500,300},
⟨ppl⟩      - = {300,500},
⟨ptm⟩      - = {500,500},
⟨ugm⟩      - = {400,700},
⟨blg⟩      _ = {0,300},
⟨m-t|pmn⟩      \textendash       = {200,200},   \textemdash        = {150,150},
⟨bch⟩      \textendash       = {200,300},   \textemdash        = {150,200},
⟨cmr⟩      \textendash       = {500,300},   \textemdash        = {400,170},
⟨ebg|ppl|ptm|ugm⟩      \textendash       = {300,300},   \textemdash        = {200,200},
⟨m-t|bch|pmn|ugm⟩      \textquoteleft    = {400,200},   \textquoteright    = {400,200},
⟨blg⟩      \textquoteleft     = {400,400},    \textquoteright    = {400,400},
⟨cmr⟩      \textquoteleft    = {800,200},   \textquoteright    = {800,-20},
⟨ebg⟩      \textquoteleft    = {800,200},   \textquoteright    = {800,200},
⟨ppl⟩      \textquoteleft    = {700,400},   \textquoteright    = {700,400},
⟨ptm⟩      \textquoteleft    = {800,500},   \textquoteright    = {800,500},
⟨m-t|bch|pmn⟩      \textquotedblleft = {400,200},   \textquotedblright = {400,200}
⟨blg⟩      \textquotedblright = {300,300}
⟨cmr⟩      \textquotedblleft = {540,100},   \textquotedblright = {500,100}
⟨ebg⟩      \textquotedblleft = {700,200},   \textquotedblright = {700,200}
⟨ppl⟩      \textquotedblleft = {500,300},   \textquotedblright = {500,300}
⟨ptm⟩      \textquotedblleft = {700,400},   \textquotedblright = {700,400}
⟨ugm⟩      \textquotedblleft = {600,200},   \textquotedblright = {600,200}
   }

⟨*cmr|ebg|pmn⟩
\SetProtrusion
⟨cmr⟩   [ name     = cmr-it-OT1,
⟨ebg⟩   [ name     = EBGaramond-it-OT1,
⟨pmn⟩   [ name     = pmnj-it-OT1,
⟨cmr⟩     load     = cmr-it   ]
⟨ebg⟩     load     = EBGaramond-it   ]
⟨pmn⟩     load     = pmnj-it  ]
⟨cmr⟩   { encoding = {OT1,OT4},
⟨pmn⟩   { encoding = OT1,
⟨cmr⟩     family   = cmr,
⟨pmn⟩     family   = pmnj,
⟨cmr⟩     shape    = it       }
⟨pmn⟩     shape    = {it,sl}  }
⟨ebg⟩   { }
   {
⟨cmr⟩     \AE = {100,   },
⟨pmn⟩     \AE = {  ,-50},
⟨cmr⟩     \OE = {100,   },
⟨pmn⟩     \OE = {50,   }
⟨*cmr|ebg⟩
⟨cmr⟩     "00 = {200,150}, % \Gamma
⟨ebg⟩     "00 = {   ,150}, % \Gamma
⟨cmr⟩     "01 = {150,100}, % \Delta
⟨ebg⟩     "01 = {100,100}, % \Delta
⟨cmr⟩     "02 = {150, 50}, % \Theta
⟨ebg⟩     "02 = { 50, 50}, % \Theta
⟨cmr⟩     "03 = {150, 50}, % \Lambda
⟨ebg⟩     "03 = {100,100}, % \Lambda
⟨cmr⟩     "04 = {100,100}, % \Xi
⟨ebg⟩     "04 = { 50, 50}, % \Xi
⟨cmr⟩     "05 = {100,100}, % \Pi
⟨cmr⟩     "06 = {100, 50}, % \Sigma
⟨cmr⟩     "07 = {200,150}, % \Upsilon
⟨ebg⟩     "07 = {100,100}, % \Upsilon
⟨cmr⟩     "08 = {150, 50}, % \Phi
⟨ebg⟩     "08 = { 50, 50}, % \Phi
⟨cmr⟩     "09 = {150,100}, % \Psi
⟨ebg⟩     "09 = { 50, 50}, % \Psi
     "0A = { 50, 50}, % \Omega
⟨ebg⟩     138 = {   , 50}, % \L
⟨/cmr|ebg⟩
   }

⟨/cmr|ebg|pmn⟩
⟨*ebg⟩
\SetProtrusion
   [ name     = EBGaramond-it-OT1-LF,
     load     = EBGaramond-it-OT1 ]
   { encoding = OT1,
     family   = {EBGaramond-LF,EBGaramond-TLF},
     shape    = it }
   {
     1 = {50,50},
     2 = {50,50},
     3 = {80,50},
     4 = {50,50},
     5 = {50,50},
     6 = {50,50},
     7 = {50,50},
     8 = {50,50},
     9 = {50,  },
   }

\SetProtrusion
   [ name     = EBGaramond-it-OT1-OsF,
     load     = EBGaramond-it-OT1 ]
   { encoding = OT1,
    family   = {EBGaramond-OsF},
    shape    = it }
  {
    1 = {50,50},
    2 = {50,50},
    3 = {  ,80},
    4 = {50,50},
    7 = {50,50},
  }

\SetProtrusion
  [ name     = EBGaramond-it-OT1-TOsF,
    load     = EBGaramond-it-OT1 ]
  { encoding = OT1,
    family   = {EBGaramond-TOsF},
    shape    = it }
  {
    0 = {150,150},
    1 = {150,150},
    2 = {80,80},
    3 = {50,80},
    4 = {50,80},
    5 = {50,80},
    6 = {50,50},
    7 = {50,100},
    8 = {50,50},
    9 = {50,80},
  }

⟨/ebg⟩
\SetProtrusion
⟨m-t⟩   [ name     = T1-it-default,
⟨bch⟩   [ name     = bch-it-T1,
⟨blg⟩   [ name     = blg-it-T1,
⟨cmr⟩   [ name     = cmr-it-T1,
⟨ebg⟩   [ name     = EBGaramond-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  ]
⟨ebg⟩     load     = EBGaramond-it   ]
⟨ppl⟩     load     = ppl-it   ]
⟨ptm⟩     load     = ptm-it   ]
⟨ugm⟩     load     = ugm-it   ]
⟨m-t|bch|cmr|pmn|ppl⟩   { encoding = {T1,LY1},
⟨ebg⟩   { encoding = {LY1},
⟨blg|ptm|ugm⟩   { encoding = T1,
⟨bch⟩     family   = bch,
⟨blg⟩     family   = blg,
⟨cmr⟩     family   = cmr,
⟨pmn⟩     family   = pmnj,
⟨ebg⟩     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
⟨ppl⟩     family   = {ppl,pplx,pplj},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
⟨ugm⟩     family   = ugm,
⟨m-t|bch|pmn|ppl|ptm⟩     shape    = {it,sl}  }
⟨blg|cmr|ebg|ugm⟩     shape    = it       }
  {
⟨m-t|bch|pmn⟩     _ = {  ,100},
⟨blg⟩     _ = {0,300},
⟨cmr|ugm⟩     _ = {100,200},
⟨ebg|ppl|ptm⟩     _ = {100,100},
⟨blg⟩     . = {400,600},
⟨blg⟩    {,}= {300,500},
⟨cmr⟩     \AE = {100,   },
⟨pmn⟩     \AE = {  ,-50},
⟨bch|pmn⟩     \OE = { 50,   },
⟨cmr⟩     \OE = {100,   },
⟨pmn⟩     031 = { ,-100}, % ffl
⟨cmr|ptm⟩     156 = {100, },  % IJ
⟨ebg⟩     156 = {50,  },  % IJ
⟨pmn⟩     156 = {20,  },  % IJ
⟨pmn⟩     188 = {  ,-30}, % ij
⟨pmn⟩   \v t = { ,100},
⟨m-t|ebg|ppl|ptm⟩     \textbackslash    = {100,200},
⟨cmr|ugm⟩     \textbackslash    = {300,300},
⟨bch⟩     \textbackslash    = {150,150},
⟨pmn⟩     \textbackslash    = {100,150},
⟨ugm⟩     \textbar          = {200,200},
⟨cmr⟩     \textquotedblleft = {500,300},
⟨blg⟩    \textquoteleft     = {400,400},    \textquoteright    = {400,400},
⟨blg⟩    \textquotedbl      = {300,300},    \textquotedblleft  = {300,300},
⟨blg⟩    \textquotedblright = {300,300},    \quotedblbase      = {200,600},
⟨m-t|ptm⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {400,500},
⟨cmr⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {200,600},
⟨bch|pmn⟩     \quotesinglbase   = {200,500},   \quotedblbase      = {150,500},
⟨ebg|ppl⟩     \quotesinglbase   = {500,500},   \quotedblbase      = {400,400},
⟨ugm⟩     \quotesinglbase   = {300,700},   \quotedblbase      = {300,500},
⟨m-t|ppl|ptm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,500},
⟨bch|pmn⟩     \guilsinglleft    = {300,400},   \guilsinglright    = {200,500},
⟨cmr⟩     \guilsinglleft    = {500,300},   \guilsinglright    = {400,400},
⟨ebg⟩     \guilsinglleft    = {500,400},   \guilsinglright    = {300,500},
⟨ugm⟩     \guilsinglleft    = {400,400},   \guilsinglright    = {300,600},
⟨m-t|ppl⟩     \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨bch|pmn⟩     \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨ebg⟩     \guillemotleft    = {300,300},   \guillemotright    = {200,400},
⟨ptm⟩     \guillemotleft    = {300,400},   \guillemotright    = {200,400},
⟨ugm⟩     \guillemotleft    = {300,400},   \guillemotright    = {300,400},
⟨m-t|ebg|ppl|ugm⟩     \textexclamdown   = {100,   },   \textquestiondown  = {200,   },
⟨cmr|ptm⟩     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
⟨pmn⟩     \textexclamdown   = {-50,   },   \textquestiondown  = {-50,   },
⟨m-t|ppl|ugm⟩     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
⟨bch|pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr|ebg|ptm⟩     \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨bch|pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100},
⟨cmr|ebg|ppl|ptm⟩     \textless         = {300,100},   \textgreater       = {200,100}
⟨pmn⟩     \textvisiblespace = {100,100}
  }

⟨*ebg⟩
\SetProtrusion
   [ name     = EBGaramond-it-T1-LF,
     load     = EBGaramond-it-T1 ]
   { encoding = T1,
     family   = {EBGaramond-LF,EBGaramond-TLF},
     shape    = it }
   {
     1 = {50,50},
     2 = {50,50},
     3 = {80,50},
     4 = {50,50},
     5 = {50,50},
     6 = {50,50},
     7 = {50,50},
     8 = {50,50},
     9 = {50,  },
   }

\SetProtrusion
   [ name     = EBGaramond-it-T1-OsF,
     load     = EBGaramond-it-T1 ]
   { encoding = T1,
     family   = {EBGaramond-OsF},
     shape    = it }
   {
     1 = {50,50},
     2 = {50,50},
     3 = {  ,80},
     4 = {50,50},
     7 = {50,50},
   }

\SetProtrusion
   [ name     = EBGaramond-it-T1-TOsF,
     load     = EBGaramond-it-T1 ]
   { encoding = T1,
     family   = {EBGaramond-TOsF},
     shape    = it }
   {
     0 = {150,150},
     1 = {150,150},
     2 = {80,80},
     3 = {50,80},
     4 = {50,80},
     5 = {50,80},
     6 = {50,50},
     7 = {50,100},
     8 = {50,50},
     9 = {50,80},
   }

⟨/ebg⟩
⟨*m-t|cmr|pmn⟩
\SetProtrusion
⟨m-t⟩   [ name     = T2A-it-default,
⟨cmr⟩   [ name     = cmr-it-T2A,
⟨pmn⟩   [ name     = pmnj-it-T2A,
⟨m-t⟩     load     = OT1-it   ]
⟨cmr⟩     load     = cmr-it   ]
⟨pmn⟩     load     = pmnj-it  ]
   { encoding = T2A,
⟨cmr⟩     family   = cmr,
⟨pmn⟩     family   = pmnj,
⟨m-t|pmn⟩     shape    = {it,sl}  }
⟨cmr⟩     shape    = it       }
   {
⟨cmr⟩     \CYRA = {100,50},
⟨pmn⟩     \CYRA = {50,  },
⟨cmr⟩     \CYRB = {50,  },
⟨cmr⟩     \CYRV = {50,  },
⟨pmn⟩     \CYRV = {20,-50},
⟨cmr⟩     \CYRG = {100, },
⟨pmn⟩     \CYRG = {10,  },
⟨cmr⟩     \CYRD = {50,  },
⟨cmr⟩     \CYRE = {50,  },
⟨pmn⟩     \CYRE = {20,-50},
⟨cmr⟩     \CYRZH = {50,  },
⟨cmr⟩     \CYRZ = {50,  },
⟨pmn⟩     \CYRZ = {20,-50},
⟨cmr⟩     \CYRI = {50,  },
⟨pmn⟩     \CYRI = {  ,-30},
⟨cmr⟩     \CYRISHRT = {50,  },
⟨cmr⟩     \CYRK = {50,  },
⟨pmn⟩     \CYRK = {20,  },
⟨cmr⟩     \CYRL = {50,  },
⟨cmr⟩     \CYRM = {50,  },
⟨pmn⟩     \CYRM = {  ,-30},
⟨cmr⟩     \CYRN = {50,  },
⟨cmr⟩     \CYRO = {100, },
⟨pmn⟩     \CYRO = {50,  },
⟨cmr⟩     \CYRP = {50,  },
⟨cmr⟩     \CYRR = {50,  },
⟨pmn⟩     \CYRR = {20,-50},
⟨cmr⟩     \CYRS = {100, },
⟨pmn⟩     \CYRS = {50,  },
⟨cmr⟩     \CYRT = {100, },
⟨pmn⟩     \CYRT = {70,  },
⟨cmr⟩     \CYRU = {100, },
⟨pmn⟩     \CYRU = {50,  },
⟨cmr⟩     \CYRF = {100, },
⟨cmr⟩     \CYRH = {50,  },
⟨cmr⟩     \CYRC = {50,  },
⟨cmr⟩     \CYRCH = {100, },
⟨cmr⟩     \CYRSH = {50,  },
⟨cmr⟩     \CYRSHCH = {50,  },
⟨cmr⟩     \CYRHRDSN = {100, },
⟨cmr⟩     \CYRERY = {50,  },
⟨cmr⟩     \CYRSFTSN = {50,  },
⟨cmr⟩     \CYREREV = {50,  },
⟨cmr⟩     \CYRYU = {50,  },
⟨cmr⟩     \CYRYA = {50,  },
⟨pmn⟩     \CYRYA = {  ,20},
⟨pmn⟩     \cyrr = {-50,  },
⟨m-t|pmn⟩      _ = {  ,100},
⟨cmr⟩      _ = {100,200},
⟨pmn⟩      031 = { ,-100}, % ffl
⟨pmn⟩     \v t = { ,100},
⟨m-t⟩     \textbackslash    = {100,200},   \quotedblbase      = {400,500},
⟨cmr⟩     \textbackslash    = {300,300},   \quotedblbase      = {200,600},
⟨pmn⟩     \textbackslash    = {100,150},   \quotedblbase      = {150,500},
⟨m-t⟩     \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨cmr⟩     \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨pmn⟩     \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨m-t⟩     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
⟨cmr⟩     \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨pmn⟩     \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr⟩     \textquotedblleft = {500,300},
⟨cmr⟩     \textless         = {300,100},   \textgreater       = {200,100}
⟨pmn⟩     \textless         = {100,   },   \textgreater       = {   ,100}
  }

⟨/m-t|cmr|pmn⟩
⟨*m-t|ptm⟩
\SetProtrusion
⟨m-t⟩   [ name     = QX-it-default,
⟨ptm⟩   [ name     = ptm-it-QX,
⟨m-t⟩     load     = OT1-it ]
⟨ptm⟩     load     = ptm-it ]
   { encoding = {QX},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
     shape    = {it,sl}  }
   {
⟨ptm⟩     009 = {  , 50}, % fk
     {=} = {100,100},
⟨m-t⟩    \textunderscore   = {100,100},
⟨ptm⟩    \textunderscore   = {100,150},
     \textbackslash    = {100,200},
     \quotedblbase     = {300,400},
⟨m-t⟩    \guillemotleft    = {300,300},   \guillemotright    = {300,300},
⟨ptm⟩    \guillemotleft    = {200,400},   \guillemotright    = {200,400},
     \textexclamdown   = {200,   },   \textquestiondown  = {200,   },
     \textbraceleft    = {200,100},   \textbraceright    = {200,200},
     \textless         = {100,100},   \textgreater       = {100,100},
     \textminus        = {200,200},   \textdegree        = {300,150},
⟨m-t⟩    \copyright        = {100,100},   \textregistered    = {100,100}
⟨ptm⟩    \textregistered   = {100,150},   \copyright         = {100,150},
⟨ptm⟩    \textDelta        = { 70,   },   \textdelta         = {   , 50},
⟨ptm⟩    \textpi           = { 50, 80},   \textmu            = {   , 80},
⟨ptm⟩    \texteuro         = {200,   },   \textellipsis      = {100,200},
⟨ptm⟩    \textquoteleft    = {500,400},   \textquoteright    = {500,400},
⟨ptm⟩    \textquotedblleft = {500,300},   \textquotedblright = {400,400},
⟨ptm⟩    \textapprox       = { 50, 50},   \textinfty         = {100,100},
⟨ptm⟩    \textdagger       = {150,150},   \textdaggerdbl     = {100,100},
⟨ptm⟩    \textdiv          = {150,150},   \textasciitilde    = { 80, 80},
⟨ptm⟩    \texttimes        = {100,150},   \textpm            = { 50, 80},
⟨ptm⟩    \textbullet       = {300,100},   \textperiodcentered = {300,300},
⟨ptm⟩    \textquotesingle  = {500,500},   \textquotedbl      = {300,300},
⟨ptm⟩    \textperthousand  = {    ,50}
   }

⟨/m-t|ptm⟩
⟨*cmr|bch⟩
\SetProtrusion
⟨cmr⟩  [ name = cmr-it-T5,
⟨cmr⟩    load = cmr-it ]
⟨bch⟩  [ name = bch-it-T5,
⟨bch⟩    load = bch-it ]
   { encoding = T5,
⟨bch⟩    family   = bch,
⟨cmr⟩    family   = cmr,
     shape    = it }
   {
⟨bch⟩     _ = {  ,100},
⟨cmr⟩     _ = {100,200},
⟨bch⟩    \textbackslash    = {150,150},
⟨cmr⟩    \textbackslash    = {300,300},
⟨bch⟩    \quotesinglbase   = {200,500},   \quotedblbase      = {150,500},
⟨cmr⟩    \quotesinglbase   = {300,700},   \quotedblbase      = {200,600},
⟨bch⟩    \guilsinglleft    = {300,400},   \guilsinglright    = {200,500},
⟨cmr⟩    \guilsinglleft    = {500,300},   \guilsinglright    = {400,400},
⟨bch⟩    \guillemotleft    = {200,300},   \guillemotright    = {150,400},
⟨cmr⟩    \guillemotleft    = {400,100},   \guillemotright    = {200,300},
⟨bch⟩    \textbraceleft    = {200,   },   \textbraceright    = {   ,200},
⟨cmr⟩    \textbraceleft    = {400,100},   \textbraceright    = {200,200},
⟨bch⟩    \textless         = {100,   },   \textgreater       = {   ,100}
⟨cmr⟩    \textless         = {300,100},   \textgreater       = {200,100}
  }

⟨/cmr|bch⟩
```

Slanted is very similar to italic.

```
⟨*cmr⟩
\SetProtrusion
   [ name     = cmr-sl,
     load     = cmr-it-OT1 ]
   { encoding = {OT1,OT4},
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T1,
     load     = cmr-it-T1 ]
   { encoding = {T1,LY1},
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T2A,
     load     = cmr-it-T2A ]
   { encoding = T2A,
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = cmr-sl-T5,
     load     = cmr-it-T5 ]
   { encoding = T5,
     family   = cmr,
     shape    = sl  }
   {
      L = {  ,50},
      f = { ,-50},
      - = {300, },
     \textendash = {400,  },  \textemdash = {300,  }
   }

\SetProtrusion
   [ name     = lmr-it-T1,
     load     = cmr-it-T1 ]
   { encoding = {T1,LY1},
     family   = lmr,
     shape    = {it,sl} }
   {
     \textquotedblleft  = { ,200},  \textquotedblright = { ,200},
     \quotesinglbase    = { ,400},  \quotedblbase      = { ,500}
   }

```

Oldstyle numerals are slightly different.

```
\SetProtrusion
   [ name = cmr(oldstyle)-it,
     load = cmr-it-T1 ]
   { encoding = T1,
     family   = {hfor,cmor},
     shape    = {it,sl}  }
   {
     1 = {250, 50},
     2 = {150,-100},
     3 = {100,-50},
     4 = {150,150},
     6 = {200,   },
     7 = {200, 50},
     8 = {150,-50},
     9 = {100, 50}
   }

⟨/cmr⟩
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-it,
     load     = pmnj-it ]
   { encoding = OT1,
     family   = pmnx,
     shape    = {it,sl} }
   {
     1 = {100,150}
   }

\SetProtrusion
   [ name     = pmnx-it-T1,
     load     = pmnj-it-T1 ]
   { encoding = {T1,LY1},
     family   = pmnx,
     shape    = {it,sl} }
   {
     1 = {100,150}
   }

\SetProtrusion
   [ name     = pmnx-it-T2A,
     load     = pmnj-it-T2A ]
   { encoding = {T2A},
     family   = pmnx,
     shape    = {it,sl} }
   {
     1 = {100,150}
   }

⟨/pmn⟩
⟨*ptm⟩
\SetProtrusion
   [ name     = ptm-it-LY1,
     load     = ptm-it-T1   ]
   { encoding = {LY1},
     family   = {ptm,ptmx,ptmj},
     shape    = {it,sl}  }
   {
     _                          = {100,100},
     \texttrademark            = {100,100},
     \textregistered           = {100,100},
     \textcopyright            = {100,100},
     \textdegree               = {300,100},
     \textminus                = {200,200},
     \textellipsis             = {100,200},
%    \texteuro                 = {   ,   }, % ?
     \textcent                 = {100,100},
     \textquotesingle          = {500,   },
     \textflorin               = {100, 70},
     \textdagger               = {150,150},
     \textdaggerdbl            = {100,100},
     \textbullet               = {150,150},
     \textonesuperior          = {150,100},
     \texttwosuperior          = {150, 50},
     \textthreesuperior        = {150, 50},
     \textparagraph            = {100,   },
     \textperiodcentered       = {500,300},
     \textonequarter           = { 50,   },
     \textonehalf              = { 50,   },
     \textplusminus            = {100,100},
     \textmultiply             = {150,150},
     \textdivide               = {150,150}
   }

⟨/ptm⟩
```

2.8.3 Small caps

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

```
⟨*!(blg|ugm)⟩
\SetProtrusion
⟨m-t⟩   [ name     = OT1-sc,
⟨bch⟩   [ name     = bch-sc,
⟨cmr⟩   [ name     = cmr-sc-OT1,
⟨ebg⟩   [ name     = EBGaramond-sc-OT1-Prop,
⟨pmn⟩   [ name     = pmnj-sc,
⟨ppl⟩   [ name     = ppl-sc,
⟨ptm⟩   [ name     = ptm-sc,
⟨m-t⟩     load     = default ]
⟨bch⟩     load     = bch-default ]
⟨cmr⟩     load     = cmr-OT1 ]
⟨ebg⟩     load     = EBGaramond-OT1-LF ]
⟨pmn⟩     load     = pmnj-default ]
⟨ppl⟩     load     = ppl-default ]
⟨ptm⟩     load     = ptm-default ]
⟨m-t|bch|ebg|pmn⟩   { encoding = OT1,
⟨cmr|ppl|ptm⟩   { encoding = {OT1,OT4},
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
⟨ebg⟩     family   = {EBGaramond-LF,EBGaramond-OsF},
⟨pmn⟩     family   = pmnj,
⟨ppl⟩     family   = {ppl,pplx,pplj},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
     shape    = sc }
   {
     a = {50,50},
⟨cmr|ebg|ppl|ptm⟩   \ae = {50,  },
⟨bch|pmn⟩     c = {50,  },
⟨bch|ebg|pmn⟩     d = {  ,50},
⟨m-t|bch|cmr|ebg|pmn|ptm⟩     f = {  ,50},
⟨bch|ebg|pmn⟩     g = {50,  },
⟨m-t|cmr|ebg|pmn|ppl|ptm⟩     j = {50,  },
⟨bch⟩     j = {100,  },
⟨m-t|bch|cmr|ebg|pmn|ppl⟩     l = {  ,50},
⟨ptm⟩     l = {  ,80},
⟨m-t|bch|cmr|pmn|ppl⟩   013 = {  ,50}, % fl
⟨ptm⟩   013 = {  ,80}, % fl
⟨bch|ebg|pmn⟩     o = {50,50},
⟨ebg|pmn⟩   \oe = {50,  },
⟨ppl⟩     p = { 0, 0},
⟨bch|ebg|pmn⟩     q = {50,70},
⟨ppl⟩     q = { 0,  },
⟨m-t|cmr|ebg|pmn|ppl|ptm⟩     r = {  , 0},
     t = {50,50},
⟨m-t|bch|cmr|ebg|pmn|ppl⟩     y = {50,50}
⟨ptm⟩     y = {80,80}
  }

⟨*ebg⟩
\SetProtrusion
   [ name     = EBGaramond-sc-OT1-Tab,
     load     = EBGaramond-OT1-TOsF ]
   { encoding = OT1,
     family   = {EBGaramond-TLF,EBGaramond-TOsF},
     shape    = sc }
   {
     a = {50,50},
   \ae = {50,  },
     d = {  ,50},
     f = {  ,50},
     g = {50,  },
     j = {50,  },
     l = {  ,50},
     o = {50,50},
   \oe = {50,  },
     q = {50,70},
     r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

⟨/ebg⟩
\SetProtrusion
⟨m-t⟩   [ name     = T1-sc,
⟨bch⟩   [ name     = bch-sc-T1,
⟨cmr⟩   [ name     = cmr-sc-T1,
⟨ebg⟩   [ name     = EBGaramond-sc-T1,
⟨pmn⟩   [ name     = pmnj-sc-T1,
⟨ppl⟩   [ name     = ppl-sc-T1,
⟨ptm⟩   [ name     = ptm-sc-T1,
⟨m-t⟩     load     = T1-default ]
⟨bch⟩     load     = bch-T1     ]
⟨cmr⟩     load     = cmr-T1     ]
⟨ebg⟩     load     = EBGaramond-T1     ]
⟨pmn⟩     load     = pmnj-T1    ]
⟨ppl⟩     load     = ppl-T1     ]
⟨ptm⟩     load     = ptm-T1     ]
⟨!ebg⟩   { encoding = {T1,LY1},
⟨ebg⟩   { encoding = {LY1},
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
⟨ebg⟩     family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
⟨pmn⟩     family   = pmnj,
⟨ppl⟩     family   = {ppl,pplx,pplj},
⟨ptm⟩     family   = {ptm,ptmx,ptmj},
     shape    = sc }
   {
     a = {50,50},
⟨cmr|ebg|ppl|ptm⟩   \ae = {50,  },
⟨bch|pmn⟩     c = {50,  },
⟨bch|ebg|pmn⟩     d = {  ,50},
⟨m-t|bch|cmr|ebg|pmn|ptm⟩     f = {  ,50},
⟨bch|ebg|pmn⟩     g = {50,  },
⟨m-t|cmr|ebg|pmn|ppl|ptm⟩     j = {50,  },
⟨bch⟩     j = {100,  },
⟨m-t|bch|cmr|ebg|pmn|ppl⟩     l = {  ,50},
⟨ptm⟩     l = {  ,80},
⟨m-t|bch|cmr|pmn|ppl⟩   029 = {  ,50}, % fl
```

```
⟨ptm⟩   029 = {  ,80}, % fl
⟨bch|ebg|pmn⟩     o = {50,50},
⟨bch|ebg|pmn⟩   \oe = {50,  },
⟨ppl⟩     p = { 0, 0},
⟨bch|ebg|pmn⟩     q = {50,70},
⟨ppl⟩     q = { 0,  },
⟨m-t|cmr|ebg|pmn|ppl|ptm⟩     r = {  , 0},
     t = {50,50},
⟨m-t|bch|cmr|ebg|pmn|ppl⟩     y = {50,50}
⟨ptm⟩     y = {80,80}
   }

⟨/!(blg|ugm)⟩
⟨*m-t|cmr⟩
\SetProtrusion
⟨m-t⟩   [ name     = T2A-sc,
⟨cmr⟩   [ name     = cmr-sc-T2A,
⟨m-t⟩     load     = T2A-default ]
⟨cmr⟩     load     = cmr-T2A     ]
   { encoding = T2A,
⟨cmr⟩     family   = cmr,
     shape    = sc }
   {
     \cyra = {50,50},
     \cyrg = {  ,50},
     \cyrt = {50,50},
     \cyry = {  ,50}
   }

⟨/m-t|cmr⟩
⟨*m-t⟩
\SetProtrusion
   [ name     = QX-sc,
     load     = QX-default ]
   { encoding = QX,
     shape    = sc  }
   {
     a = {50,50},
     f = {  ,50},
     j = {50,  },
     l = {  ,50},
   013 = {  ,50}, % fl
     r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

⟨/m-t⟩
⟨*cmr|bch⟩
\SetProtrusion
⟨bch⟩   [ name     = bch-sc-T5,
⟨bch⟩     load     = bch-T5 ]
⟨cmr⟩   [ name     = cmr-sc-T5,
⟨cmr⟩     load     = cmr-T5 ]
   { encoding = T5,
⟨bch⟩     family   = bch,
⟨cmr⟩     family   = cmr,
     shape    = sc }
   {
     a = {50,50},
⟨bch⟩     c = {50,  },
⟨bch⟩     d = {  ,50},
     f = {  ,50},
⟨bch⟩     g = {50,  },
⟨bch⟩     j = {100, },
⟨cmr⟩      j = {50,  },
     l = {  ,50},
⟨bch⟩      o = {50,50},
⟨bch⟩      q = { 0,  },
⟨cmr⟩      r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

⟨/cmr|bch⟩
⟨*ebg⟩
\SetProtrusion
   [ name     = EBGaramond-sc-T1-Prop,
     load     = EBGaramond-T1-LF ]
   { encoding = T1,
     family   = {EBGaramond-LF,EBGaramond-OsF},
     shape    = sc }
   {
     a = {50,50},
   \ae = {50,  },
     d = {  ,50},
     f = {  ,50},
     g = {50,  },
     j = {50,  },
     l = {  ,50},
     o = {50,50},
   \oe = {50,  },
     q = {50,70},
     r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

\SetProtrusion
   [ name     = EBGaramond-sc-T1-Tab,
     load     = EBGaramond-T1-TOsF ]
   { encoding = T1,
     family   = {EBGaramond-TLF,EBGaramond-TOsF},
     shape    = sc }
   {
     a = {50,50},
   \ae = {50,  },
     d = {  ,50},
     f = {  ,50},
     g = {50,  },
     j = {50,  },
     l = {  ,50},
     o = {50,50},
   \oe = {50,  },
     q = {50,70},
     r = {  , 0},
     t = {50,50},
     y = {50,50}
   }

⟨/ebg⟩
⟨*pmn⟩
\SetProtrusion
   [ name     = pmnx-sc,
     load     = pmnj-sc ]
   { encoding = OT1,
     family   = pmnx,
     shape    = sc }
   {
     1 = {230,180}
```

```
  }

\SetProtrusion
  [ name     = pmnx-sc-T1,
    load     = pmnj-sc-T1 ]
  { encoding = {T1,LY1},
    family   = pmnx,
    shape    = sc }
  {
    1 = {230,180}
  }

```

2.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```
\SetProtrusion
  [ name     = pmnj-scit,
    load     = pmnj-it   ]
  { encoding = OT1,
    family   = pmnj,
    shape    = {scit,si} }
  {
    a = {50,  },
  \ae = {  ,-50},
    b = {20,-50},
    c = {50,-50},
    d = {20, 0},
    e = {20,-50},
    f = {10, 0},
  012 = {10,-50}, % fi
  013 = {10,-50}, % fl
  014 = {10,-50}, % ffi
  015 = {10,-50}, % ffl
    g = {50,-50},
    i = {20,-50},
    j = {20, 0},
    k = {20,  },
    l = {20,50},
    m = {  ,-30},
    n = {  ,-30},
    o = {50,  },
  \oe = {50,-50},
    p = {20,-50},
    q = {50,  },
    r = {20, 0},
    s = {20,-30},
    t = {70,  },
    u = {50,-50},
    v = {100,  },
    w = {100,  },
    y = {50,  },
    z = {  ,-50}
  }

\SetProtrusion
  [ name     = pmnj-scit-T1,
    load     = pmnj-it-T1   ]
  { encoding = {T1,LY1},
    family   = pmnj,
    shape    = {scit,si}    }
  {
    a = {50,  },
  \ae = {  ,-50},
    b = {20,-50},
    c = {50,-50},
    d = {20, 0},
    e = {20,-50},
    f = {10, 0},
  028 = {10,-50}, % fi
  029 = {10,-50}, % fl
  030 = {10,-50}, % ffi
  031 = {10,-50}, % ffl
    g = {50,-50},
    i = {20,-50},
  188 = {20, 0},  % ij
    j = {20, 0},
    k = {20,  },
    l = {20,50},
    m = {  ,-30},
    n = {  ,-30},
    o = {50,  },
  \oe = {50,-50},
    p = {20,-50},
    q = {50,  },
    r = {20, 0},
    s = {20,-30},
    t = {70,  },
    u = {50,-50},
    v = {100,  },
    w = {100,  },
    y = {50,  },
    z = {  ,-50}
  }

\SetProtrusion
  [ name     = pmnx-scit,
    load     = pmnj-scit ]
  { encoding = OT1,
    family   = pmnx,
    shape    = {scit,si} }
  {
    1 = {100,150}
  }

\SetProtrusion
  [ name     = pmnx-scit-T1,
    load     = pmnj-scit-T1 ]
  { encoding = {T1,LY1},
    family   = pmnx,
    shape    = {scit,si}    }
  {
    1 = {100,150}
  }

⟨/pmn⟩
⟨*ebg⟩
```

For small caps italics, we copy the definitions from the small caps settings, except that we first load the italics settings.

```
\SetProtrusion
  [ name     = EBGaramond-scit-OT1-Prop,
    load     = EBGaramond-it-OT1-LF ]
  { encoding = OT1,
    family   = {EBGaramond-LF,EBGaramond-OsF},
    shape    = scit }
  {
    a = {50,50},
  \ae = {50,  },
    d = {  ,50},
    f = {  ,50},
    g = {50,  },
    j = {50,  },
    l = {  ,50},
    o = {50,50},
  \oe = {50,  },
    q = {50,70},
    r = {  , 0},
    t = {50,50},
    y = {50,50}
  }

\SetProtrusion
  [ name     = EBGaramond-scit-OT1-Tab,
    load     = EBGaramond-it-OT1-TOsF ]
  { encoding = OT1,
    family   = {EBGaramond-TLF,EBGaramond-TOsF},
    shape    = scit }
  {
    a = {50,50},
  \ae = {50,  },
    d = {  ,50},
    f = {  ,50},
    g = {50,  },
    j = {50,  },
    l = {  ,50},
    o = {50,50},
  \oe = {50,  },
    q = {50,70},
    r = {  , 0},
    t = {50,50},
    y = {50,50}
  }

\SetProtrusion
  [ name     = EBGaramond-scit-T1-Prop,
    load     = EBGaramond-it-T1-LF ]
  { encoding = T1,
    family   = {EBGaramond-LF,EBGaramond-OsF},
    shape    = scit }
  {
    a = {50,50},
  \ae = {50,  },
    d = {  ,50},
    f = {  ,50},
    g = {50,  },
    j = {50,  },
    l = {  ,50},
    o = {50,50},
  \oe = {50,  },
    q = {50,70},
    r = {  , 0},
    t = {50,50},
    y = {50,50}
  }

\SetProtrusion
  [ name     = EBGaramond-scit-T1-Tab,
    load     = EBGaramond-it-T1-TOsF ]
  { encoding = T1,
    family   = {EBGaramond-TLF,EBGaramond-TOsF},
    shape    = scit }
  {
    a = {50,50},
  \ae = {50,  },
    d = {  ,50},
    f = {  ,50},
    g = {50,  },
    j = {50,  },
    l = {  ,50},
    o = {50,50},
  \oe = {50,  },
    q = {50,70},
    r = {  , 0},
    t = {50,50},
    y = {50,50}
  }

⟨/ebg⟩
```

2.8.5 Text companion

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

```
\SetProtrusion
⟨m-t⟩    [ name     = textcomp ]
⟨bch⟩    [ name     = bch-textcomp ]
⟨blg⟩    [ name     = blg-textcomp ]
⟨cmr⟩    [ name     = cmr-textcomp ]
⟨ebg⟩    [ name     = EBGaramond-textcomp ]
⟨pmn⟩    [ name     = pmn-textcomp ]
⟨ppl⟩    [ name     = ppl-textcomp ]
⟨ptm⟩    [ name     = ptm-textcomp ]
⟨ugm⟩    [ name     = ugm-textcomp ]
⟨m-t⟩    { encoding = TS1      }
⟨!m-t⟩    { encoding = TS1,
⟨bch⟩      family   = bch }
⟨blg⟩      family   = blg }
⟨cmr⟩      family   = cmr }
⟨ebg⟩      family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF} }
⟨pmn⟩      family   = {pmnx,pmnj} }
⟨ppl⟩      family   = {ppl,pplx,pplj} }
⟨ptm⟩      family   = {ptm,ptmx,ptmj} }
⟨ugm⟩      family   = ugm }
    {
⟨blg⟩      \textquotestraightbase    = {400,500},
⟨cmr⟩      \textquotestraightbase    = {300,300},
⟨ebg|pmn⟩      \textquotestraightbase    = {400,400},
⟨blg⟩      \textquotestraightdblbase = {300,400},
⟨cmr|pmn⟩      \textquotestraightdblbase = {300,300},
⟨ebg⟩      \textquotestraightdblbase = {400,400},
⟨bch|cmr|ebg|pmn|ugm⟩      \texttwelveudash          = {200,200},
⟨bch|cmr|ebg|pmn⟩      \textthreequartersemdash  = {150,150},
⟨ugm⟩      \textthreequartersemdash  = {200,200},
⟨blg⟩      \textquotesingle          = {500,600},
⟨cmr|pmn⟩      \textquotesingle          = {300,400},
⟨ebg⟩      \textquotesingle          = {400,500},
⟨ptm⟩      \textquotesingle          = {500,500},
⟨ugm⟩      \textquotesingle          = {300,500},
⟨bch|cmr|pmn⟩      \textasteriskcentered     = {200,300},
⟨blg⟩      \textasteriskcentered     = {150,200},
⟨ebg⟩      \textasteriskcentered     = {300,300},
⟨ugm⟩      \textasteriskcentered     = {100,200},
⟨pmn⟩      \textfractionsolidus      = {-200,-200},
⟨cmr⟩      \textoneoldstyle          = {100,100},
⟨pmn⟩      \textoneoldstyle          = {   , 50},
⟨cmr⟩      \textthreeoldstyle        = {   , 50},
⟨ebg|pmn⟩    \textthreeoldstyle        = { 50,   },
⟨cmr⟩    \textfouroldstyle         = { 50, 50},
⟨ebg|pmn⟩    \textfouroldstyle         = { 50,   },
⟨cmr|ebg|pmn⟩    \textsevenoldstyle        = { 50, 80},
⟨cmr⟩    \textlangle               = {400,   },
⟨cmr⟩    \textrangle               = {   ,400},
⟨m-t|bch|pmn|ptm⟩    \textminus                = {200,200},
⟨cmr|ebg|ppl⟩    \textminus                = {300,300},
⟨blg|ugm⟩    \textminus                = {250,300},
⟨bch|ebg|pmn⟩    \textlbrackdbl            = {100,   },
⟨blg⟩    \textlbrackdbl            = {200,   },
⟨bch|ebg|pmn⟩    \textrbrackdbl            = {   ,100},
⟨blg⟩    \textrbrackdbl            = {   ,200},
⟨pmn⟩    \textasciigrave           = {200,500},
⟨bch|blg|cmr|ebg|pmn⟩    \texttildelow             = {200,250},
⟨pmn⟩    \textasciibreve           = {300,400},
⟨pmn⟩    \textasciicaron           = {300,400},
⟨pmn⟩    \textacutedbl             = {200,300},
⟨pmn⟩    \textgravedbl             = {150,300},
⟨bch|pmn|ugm⟩    \textdagger               = { 80, 80},
⟨blg⟩    \textdagger               = {200,200},
⟨cmr|ebg⟩    \textdagger               = {100,100},
⟨ptm⟩    \textdagger               = {150,150},
⟨blg⟩    \textdaggerdbl            = {150,150},
⟨cmr|ebg|pmn⟩    \textdaggerdbl            = { 80, 80},
⟨ptm⟩    \textdaggerdbl            = {100,100},
⟨bch⟩    \textbardbl               = {100,100},
⟨blg|ugm⟩    \textbardbl               = {150,150},
⟨bch⟩    \textbullet               = {200,200},
⟨blg⟩    \textbullet               = {400,500},
⟨cmr|ebg|pmn⟩    \textbullet               = {   ,100},
⟨ptm⟩    \textbullet               = {150,150},
⟨ugm⟩    \textbullet               = { 50,100},
⟨bch|cmr|pmn⟩    \textcelsius              = { 50,   },
⟨ebg⟩    \textcelsius              = { 80,   },
⟨bch⟩    \textflorin               = { 50, 50},
⟨blg⟩    \textflorin               = {100,100},
⟨ebg|ugm⟩    \textflorin               = {   ,100},
⟨pmn⟩    \textflorin               = { 50,100},
⟨ptm⟩    \textflorin               = { 50, 70},
⟨cmr⟩    \textcolonmonetary        = {   , 50},
⟨ebg|pmn⟩    \textcolonmonetary        = { 50,   },
⟨pmn⟩    \textinterrobang          = {   ,100},
⟨pmn⟩    \textinterrobangdown      = {100,   },
⟨m-t|ebg|ptm⟩    \texttrademark            = {100,100},
⟨bch⟩    \texttrademark            = {150,150},
⟨blg|cmr|ppl⟩    \texttrademark            = {200,200},
⟨pmn⟩    \texttrademark            = { 50, 50},
⟨ugm⟩    \texttrademark            = {100,150},
⟨bch|ugm⟩    \textcent                 = { 50,   },
⟨ptm⟩    \textcent                 = {100,100},
⟨bch⟩    \textsterling             = { 50,   },
⟨ugm⟩    \textsterling             = {   , 50},
⟨bch⟩    \textbrokenbar            = {200,200},
⟨blg⟩    \textbrokenbar            = {250,250},
⟨ugm⟩    \textbrokenbar            = {200,300},
⟨pmn⟩    \textasciidieresis        = {300,400},
⟨m-t|bch|cmr|ebg|ptm|ugm⟩    \textcopyright            = {100,100},
⟨pmn⟩    \textcopyright            = {100,150},
⟨ppl⟩    \textcopyright            = {200,200},
⟨bch|cmr|ugm⟩    \textordfeminine          = {100,200},
⟨ebg|pmn⟩    \textordfeminine          = {200,200},
⟨bch|cmr|ebg|pmn|ugm⟩    \textlnot                 = {200,   },
⟨blg⟩    \textlnot                 = {200,100},
⟨m-t|bch|cmr|ebg|ptm|ugm⟩    \textregistered           = {100,100},
⟨pmn⟩      \textregistered           = { 50,150},
⟨ppl⟩      \textregistered           = {200,200},
⟨pmn⟩      \textasciimacron          = {150,200},
⟨m-t|ppl|ptm⟩      \textdegree               = {300,300},
⟨bch⟩      \textdegree               = {150,200},
⟨blg|ugm⟩      \textdegree               = {200,200},
⟨cmr|ebg⟩      \textdegree               = {400,400},
⟨pmn⟩      \textdegree               = {150,400},
⟨bch|cmr|ebg|pmn|ugm⟩      \textpm                   = {150,200},
⟨blg⟩      \textpm                   = {100,100},
⟨ptm⟩      \textpm                   = { 50, 80},
⟨bch|blg|ugm⟩      \texttwosuperior          = {100,200},
⟨cmr⟩      \texttwosuperior          = { 50,100},
⟨ebg|pmn⟩      \texttwosuperior          = {200,200},
⟨ptm⟩      \texttwosuperior          = { 50, 50},
⟨bch|blg|ugm⟩      \textthreesuperior        = {100,200},
⟨cmr⟩      \textthreesuperior        = { 50,100},
⟨ebg|pmn⟩      \textthreesuperior        = {200,200},
⟨ptm⟩      \textthreesuperior        = { 50, 50},
⟨pmn⟩      \textasciiacute           = {300,400},
⟨bch|ugm⟩      \textmu                   = {   ,100},
⟨bch|ebg|pmn⟩      \textparagraph            = {   ,100},
⟨bch|cmr|ebg|pmn⟩      \textperiodcentered       = {300,400},
⟨blg⟩      \textperiodcentered       = {400,500},
⟨ptm⟩      \textperiodcentered       = {300,300},
⟨ugm⟩      \textperiodcentered       = {200,500},
⟨bch|blg|ugm⟩      \textonesuperior          = {200,300},
⟨cmr|ebg|pmn⟩      \textonesuperior          = {200,200},
⟨ptm⟩      \textonesuperior          = {100,100},
⟨bch|ebg|pmn|ugm⟩      \textordmasculine         = {200,200},
⟨blg|cmr⟩      \textordmasculine         = {100,200},
⟨bch|cmr|pmn⟩      \texteuro                 = {100,   },
⟨ebg⟩      \texteuro                 = { 50,100},
⟨bch⟩      \texttimes                = {200,200},
⟨blg|ptm⟩      \texttimes                = {100,100},
⟨cmr⟩      \texttimes                = {150,250},
⟨ebg⟩      \texttimes                = {100,150},
⟨pmn⟩      \texttimes                = { 70,100},
⟨ugm⟩      \texttimes                = {200,300},
⟨bch|ebg|pmn⟩      \textdiv                  = {150,200}
⟨blg⟩      \textdiv                  = {100,100}
⟨cmr⟩      \textdiv                  = {150,250}
⟨ptm⟩      \textdiv                  = { 50,100},
⟨ugm⟩      \textdiv                  = {200,300},
⟨ptm⟩      \textperthousand          = {    ,50}
⟨ugm⟩      \textsection              = {   ,100},
⟨ugm⟩      \textonehalf              = { 50,100},
⟨ugm⟩      \textonequarter           = { 50,100},
⟨ugm⟩      \textthreequarters        = { 50,100},
⟨ugm⟩      \textsurd                 = {   ,100}
```

Remaining slots in the source file.

```
   }

⟨*cmr|ebg|pmn|ugm⟩
\SetProtrusion
⟨cmr⟩    [ name     = cmr-textcomp-it ]
⟨ebg⟩    [ name     = EBGaramond-textcomp-it ]
⟨pmn⟩    [ name     = pmn-textcomp-it ]
⟨ugm⟩    [ name     = ugm-textcomp-it ]
   { encoding = TS1,
⟨cmr⟩      family   = cmr,
⟨ebg⟩      family   = {EBGaramond-LF,EBGaramond-TLF,EBGaramond-OsF,EBGaramond-TOsF},
⟨pmn⟩      family   = {pmnx,pmnj},
⟨ugm⟩      family   = ugm,
⟨cmr|pmn⟩     shape    = {it,sl} }
⟨ebg|ugm⟩     shape    = it }
    {
⟨cmr⟩     \textquotestraightbase    = {300,600},
⟨ebg|pmn⟩     \textquotestraightbase    = {400,400},
⟨cmr⟩     \textquotestraightdblbase = {300,600},
⟨ebg⟩     \textquotestraightdblbase = {300,400},
⟨pmn⟩     \textquotestraightdblbase = {300,300},
     \texttwelveudash          = {200,200},
⟨cmr|ebg|pmn⟩     \textthreequartersemdash  = {150,150},
⟨ugm⟩     \textthreequartersemdash  = {200,200},
⟨cmr⟩     \textquotesingle          = {600,300},
⟨ebg⟩     \textquotesingle          = {800,100},
⟨pmn⟩     \textquotesingle          = {300,200},
⟨ugm⟩     \textquotesingle          = {500,500},
⟨cmr⟩     \textasteriskcentered     = {300,200},
⟨ebg⟩     \textasteriskcentered     = {500,100},
⟨pmn⟩     \textasteriskcentered     = {200,300},
⟨ugm⟩     \textasteriskcentered     = {300,150},
⟨pmn⟩     \textfractionsolidus      = {-200,-200},
⟨cmr⟩     \textoneoldstyle          = {100, 50},
⟨ebg⟩     \textoneoldstyle          = {100,   },
⟨pmn⟩     \textoneoldstyle          = { 50,   },
⟨ebg⟩     \texttwooldstyle          = { 50,   },
⟨pmn⟩     \texttwooldstyle          = {-50,   },
⟨cmr⟩     \textthreeoldstyle        = {100, 50},
⟨pmn⟩     \textthreeoldstyle        = {-100,  },
⟨cmr⟩     \textfouroldstyle         = { 50, 50},
⟨ebg⟩     \textfouroldstyle         = { 50,100},
⟨cmr⟩     \textsevenoldstyle        = { 50, 80},
⟨ebg⟩     \textsevenoldstyle        = { 50,   },
⟨pmn⟩     \textsevenoldstyle        = { 20,   },
⟨cmr⟩     \textlangle               = {400,   },
⟨cmr⟩     \textrangle               = {   ,400},
⟨cmr|ebg⟩     \textminus                = {300,300},
⟨pmn⟩     \textminus                = {200,200},
⟨ugm⟩     \textminus                = {250,300},
⟨ebg|pmn⟩     \textlbrackdbl            = {100,   },
⟨ebg|pmn⟩     \textrbrackdbl            = {   ,100},
⟨pmn⟩     \textasciigrave           = {300,300},
⟨cmr|ebg|pmn⟩     \texttildelow             = {200,250},
⟨pmn⟩     \textasciibreve           = {300,300},
⟨pmn⟩     \textasciicaron           = {300,300},
⟨pmn⟩     \textacutedbl             = {200,300},
⟨pmn⟩     \textgravedbl             = {150,300},
⟨cmr⟩     \textdagger               = {100,100},
⟨ebg⟩     \textdagger               = {200,100},
⟨pmn⟩     \textdagger               = { 80, 50},
⟨ugm⟩     \textdagger               = { 80, 80},
⟨cmr|ebg⟩     \textdaggerdbl            = { 80, 80},
⟨pmn⟩     \textdaggerdbl            = { 80, 50},
⟨ugm⟩     \textbardbl               = {150,150},
⟨cmr⟩     \textbullet               = {200,100},
⟨ebg⟩     \textbullet               = {300,   },
⟨pmn⟩     \textbullet               = { 30, 70},
⟨ugm⟩     \textbullet               = { 50,100},
⟨cmr⟩     \textcelsius              = {100,   },
⟨ebg⟩     \textcelsius              = {200,   },
⟨pmn⟩     \textcelsius              = { 50,-50},
⟨ebg⟩     \textflorin               = {100,   },
⟨pmn⟩     \textflorin               = { 50,100},
⟨ugm⟩     \textflorin               = {   ,100},
⟨cmr⟩     \textcolonmonetary        = {150,   },
⟨ebg⟩     \textcolonmonetary        = {100,   },
⟨pmn⟩     \textcolonmonetary        = { 50,-50},
⟨cmr|ebg⟩     \texttrademark            = {200,   },
⟨pmn⟩     \texttrademark            = { 50,100},
⟨ugm⟩     \texttrademark            = {150, 50},
⟨ugm⟩     \textcent                 = { 50,   },
⟨ugm⟩     \textsterling             = {   , 50},
⟨ugm⟩     \textbrokenbar            = {200,300},
⟨pmn⟩     \textasciidieresis        = {300,200},
⟨cmr⟩     \textcopyright            = {100,   },
⟨ebg⟩     \textcopyright            = {200,100},
⟨pmn⟩     \textcopyright            = {100,150},
⟨ugm⟩     \textcopyright            = {300,   },
⟨cmr⟩     \textordfeminine          = {100,100},
⟨pmn⟩     \textordfeminine          = {200,200},
⟨ugm⟩     \textordfeminine          = {100,200},
⟨cmr|ebg⟩     \textlnot                 = {300,   },
⟨pmn|ugm⟩     \textlnot                 = {200,   },
⟨cmr⟩     \textregistered           = {100,   },
⟨ebg⟩     \textregistered           = {200,100},
⟨pmn⟩     \textregistered           = { 50,150},
⟨ugm⟩     \textregistered           = {300,   },
⟨pmn⟩     \textasciimacron          = {150,200},
⟨cmr|ebg⟩     \textdegree               = {500,100},
⟨pmn⟩     \textdegree               = {150,150},
⟨ugm⟩     \textdegree               = {300,200},
⟨cmr⟩     \textpm                   = {150,100},
⟨ebg⟩     \textpm                   = {200,150},
⟨pmn|ugm⟩     \textpm                   = {150,200},
⟨cmr⟩     \textonesuperior          = {400,   },
⟨ebg⟩     \textonesuperior          = {300,100},
⟨pmn⟩     \textonesuperior          = {200,100},
⟨ugm⟩     \textonesuperior          = {300,300},
⟨cmr⟩     \texttwosuperior          = {400,   },
⟨ebg⟩     \texttwosuperior          = {300,   },
⟨pmn⟩     \texttwosuperior          = {200,100},
⟨ugm⟩     \texttwosuperior          = {300,200},
⟨cmr⟩     \textthreesuperior        = {400,   },
⟨ebg⟩     \textthreesuperior        = {300,   },
⟨pmn⟩     \textthreesuperior        = {200,100},
⟨ugm⟩     \textthreesuperior        = {300,200},
⟨ugm⟩     \textmu                   = {   ,100},
⟨pmn⟩     \textasciiacute           = {300,200},
⟨cmr⟩     \textparagraph            = {200,   },
⟨pmn⟩     \textparagraph            = {   ,100},
⟨cmr⟩     \textperiodcentered       = {500,500},
⟨ebg|pmn|ugm⟩     \textperiodcentered       = {300,400},
⟨cmr⟩     \textordmasculine         = {100,100},
⟨pmn⟩     \textordmasculine         = {200,200},
⟨ugm⟩     \textordmasculine         = {300,200},
⟨cmr⟩     \texteuro                 = {200,   },
⟨ebg⟩     \texteuro                 = {100,   },
⟨pmn⟩     \texteuro                 = {100,-50},
⟨cmr⟩     \texttimes                = {200,200},
⟨ebg⟩     \texttimes                = {200,100},
⟨pmn⟩     \texttimes                = { 70,100},
⟨ugm⟩     \texttimes                = {200,300},
⟨cmr|ebg⟩     \textdiv                  = {200,200}
⟨pmn⟩     \textdiv                  = {150,200}
⟨ugm⟩     \textdiv                  = {200,300},
⟨ugm⟩     \textsection              = {   ,200},
⟨ugm⟩     \textonehalf              = { 50,100},
⟨ugm⟩     \textonequarter           = { 50,100},
⟨ugm⟩     \textthreequarters        = { 50,100},
⟨ugm⟩     \textsurd                 = {   ,100}
   }

```

```
⟨/cmr|ebg|pmn|ugm⟩
```

2.8.6 Computer Modern math

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

The math font '`operators`' (also used for the `\mathrm` and `\mathbf` alphabets) is `OT1/cmr`, which we've already set up above. It's declared as:

```
\DeclareSymbolFont{operators}  {OT1}{cmr}{m} {n}
\SetSymbolFont{operators}{bold}{OT1}{cmr}{bx}{n}
```

`\mathit` (`OT1/cmr/m/it`) is also already set up.

There are (for the moment) no settings for `\mathsf` and `\mathtt`.

Math font '`letters`' (also used as `\mathnormal`) is declared as:

```
\DeclareSymbolFont{letters}    {OML}{cmm}{m}{it}
\SetSymbolFont{letters}  {bold}{OML}{cmm}{b}{it}
```

```
⟨*cmr⟩
\SetProtrusion
   [ name     = cmr-math-letters ]
   { encoding = OML,
     family   = cmm,
     series   = {m,b},
     shape    = it    }
   {
       A = {100, 50}, % \mathnormal
       B = { 50,   },
       C = { 50,   },
       D = { 50, 50},
       E = { 50,   },
       F = {100, 50},
       G = { 50, 50},
       H = { 50, 50},
       I = { 50, 50},
       J = {150, 50},
       K = { 50,100},
       L = { 50, 50},
       M = { 50,   },
       N = { 50,   },
       O = { 50,   },
       P = { 50,   },
       Q = { 50, 50},
       R = { 50,   },
       S = { 50,   },
       T = { 50,100},
       U = { 50, 50},
       V = {100,100},
       W = { 50,100},
       X = { 50,100},
       Y = {100,100},
       f = {100,100},
       h = {   ,100},
       i = {   , 50},
       j = {   , 50},
       k = {   , 50},
       r = {   , 50},
       v = {   , 50},
       w = {   , 50},
       x = {   , 50},
     "OB = { 50,100}, % \alpha
     "0C = { 50, 50}, % \beta
     "0D = {200,150}, % \gamma
     "0E = { 50, 50}, % \delta
     "0F = { 50, 50}, % \epsilon
     "10 = { 50,150}, % \zeta
     "12 = { 50,   }, % \theta
     "13 = {   ,100}, % \iota
     "14 = {   ,100}, % \kappa
     "15 = {100, 50}, % \lambda
     "16 = {   , 50}, % \mu
     "17 = {   , 50}, % \nu
     "18 = {   , 50}, % \xi
     "19 = { 50,100}, % \pi
     "1A = { 50, 50}, % \rho
     "1B = {   ,150}, % \sigma
     "1C = { 50,150}, % \tau
     "1D = { 50, 50}, % \upsilon
     "1F = { 50,100}, % \chi
     "20 = { 50, 50}, % \psi
     "21 = {   , 50}, % \omega
     "22 = {   , 50}, % \varepsilon
     "23 = {   , 50}, % \vartheta
     "24 = {   , 50}, % \varpi
     "25 = {100,   }, % \varrho
     "26 = {100,100}, % \varsigma
     "27 = { 50, 50}, % \varphi
     "28 = {100,100}, % \leftharpoonup
     "29 = {100,100}, % \leftharpoondown
     "2A = {100,100}, % \rightharpoonup
     "2B = {100,100}, % \rightharpoondown
     "2C = {300,200}, % \lhook
     "2D = {200,300}, % \rhook
     "2E = {   ,100}, % \triangleright
     "2F = {100,   }, % \triangleleft
     "3A = {   ,500}, % ., \ldotp
     "3B = {   ,500}, % ,
     "3C = {200,100}, % <
     "3D = {300,400}, % /
     "3E = {100,200}, % >
     "3F = {200,200}, % \star
     "5B = {   ,100}, % \flat
     "5E = {200,200}, % \smile
     "5F = {200,200}, % \frown
     "7C = {100,   }, % \jmath
     "7D = {   ,100}  % \wp
```

Remaining slots in the source file.

```
   }

```

Math font ‘symbols’ (also used for the `\mathcal` alphabet) is declared as:

```
\DeclareSymbolFont{symbols}    {OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols}  {bold}{OMS}{cmsy}{b}{n}
```

```
\SetProtrusion
   [ name     = cmr-math-symbols ]
   { encoding = OMS,
     family   = cmsy,
     series   = {m,b},
     shape    = n  }
   {
       A = {150, 50}, % \mathcal
       C = {   ,100},
       D = {   , 50},
       F = { 50,150},
       I = {   ,100},
       J = {100,150},
       K = {   ,100},
       L = {100,   },
       M = { 50, 50},
       N = { 50,100},
       P = {   , 50},
       Q = { 50,   },
       R = {   , 50},
       T = { 50,150},
       V = { 50, 50},
       W = {   , 50},
       X = {100,100},
       Y = {100,   },
       Z = {100,150},
     "00 = {300,300}, % -
     "01 = {   ,700}, % \cdot, \cdotp
     "02 = {150,250}, % \times
     "03 = {150,250}, % *, \ast
     "04 = {200,300}, % \div
     "05 = {150,250}, % \diamond
     "06 = {200,200}, % \pm
     "07 = {200,200}, % \mp
     "08 = {100,100}, % \oplus
     "09 = {100,100}, % \ominus
     "0A = {100,100}, % \otimes
     "0B = {100,100}, % \oslash
     "0C = {100,100}, % \odot
     "0D = {100,100}, % \bigcirc
     "0E = {100,100}, % \circ
     "0F = {100,100}, % \bullet
     "10 = {100,100}, % \asymp
     "11 = {100,100}, % \equiv
     "12 = {200,100}, % \subseteq
     "13 = {100,200}, % \supseteq
     "14 = {200,100}, % \leq
     "15 = {100,200}, % \geq
     "16 = {200,100}, % \preceq
     "17 = {100,200}, % \succeq
     "18 = {200,200}, % \sim
     "19 = {150,150}, % \approx
     "1A = {200,100}, % \subset
     "1B = {100,200}, % \supset
     "1C = {200,100}, % \ll
     "1D = {100,200}, % \gg
     "1E = {300,100}, % \prec
     "1F = {100,300}, % \succ
     "20 = {100,200}, % \leftarrow
     "21 = {200,100}, % \rightarrow
     "22 = {100,100}, % \uparrow
     "23 = {100,100}, % \downarrow
     "24 = {100,100}, % \leftrightarrow
     "25 = {100,100}, % \nearrow
     "26 = {100,100}, % \searrow
     "27 = {100,100}, % \simeq
     "28 = {100,100}, % \Leftarrow
     "29 = {100,100}, % \Rightarrow
     "2A = {100,100}, % \Uparrow
     "2B = {100,100}, % \Downarrow
     "2C = {100,100}, % \Leftrightarrow
     "2D = {100,100}, % \nwarrow
     "2E = {100,100}, % \swarrow
     "2F = {   ,100}, % \propto
     "30 = {   ,400}, % \prime
     "31 = {100,100}, % \infty
     "32 = {150,100}, % \in
     "33 = {100,150}, % \ni
     "34 = {100,100}, % \triangle, \bigtriangleup
     "35 = {100,100}, % \bigtriangledown
     "38 = {   ,100}, % \forall
     "39 = {100,   }, % \exists
     "3A = {200,   }, % \neg
     "3E = {200,200}, % \top
     "3F = {200,200}, % \bot, \perp
     "5E = {100,200}, % \wedge
     "5F = {100,200}, % \vee
     "60 = {   ,300}, % \vdash
     "61 = {300,   }, % \dashv
     "62 = {100,100}, % \lfloor
     "63 = {100,100}, % \rfloor
     "64 = {100,100}, % \lceil
     "65 = {100,100}, % \rceil
     "66 = {150,   }, % \lbrace
     "67 = {   ,150}, % \rbrace
     "68 = {400,   }, % \langle
     "69 = {   ,400}, % \rangle
     "6C = {100,100}, % \updownarrow
     "6D = {100,100}, % \Updownarrow
     "6E = {100,300}, % \, \backslash, \setminus
     "72 = {100,100}, % \nabla
     "79 = {200,200}, % \dagger
     "7A = {100,100}, % \ddagger
     "7B = {100,   }, % \mathparagraph
     "7C = {100,100}, % \clubsuit
     "7D = {100,100}, % \diamondsuit
     "7E = {100,100}, % \heartsuit
     "7F = {100,100}  % \spadesuit
```

Remaining slots in the source file.

```
  }

```

We don’t bother about ‘largesymbols’, since it will only be used in display math, where protrusion doesn’t work anyway. It’s declared as:

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

```
⟨/cmr⟩
⟨/cfg-t⟩
```

2.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
⟨*cfg-u⟩
```

Symbol font ‘a’.

```
⟨*msa⟩
\SetProtrusion
  [ name     = AMS-a ]
  { encoding = U,
    family   = msa  }
  {
    "05  =  {150,250},  % \centerdot
    "06  =  {100,100},  % \lozenge
    "07  =  { 50, 50},  % \blacklozenge
    "08  =  { 50, 50},  % \circlearrowright
    "09  =  { 50, 50},  % \circlearrowleft
    "0A  =  {100,100},  % \rightleftharpoons
    "0B  =  {100,100},  % \leftrightharpoons
    "0D  =  {-50,200},  % \Vdash
    "0E  =  {-50,200},  % \Vvdash
    "0F  =  {-70,150},  % \vDash
    "10  =  {100,150},  % \twoheadrightarrow
    "11  =  {100,150},  % \twoheadleftarrow
    "12  =  { 50,100},  % \leftleftarrows
    "13  =  { 50, 80},  % \rightrightarrows
    "14  =  {120,120},  % \upuparrows
    "15  =  {120,120},  % \downdownarrows
    "16  =  {200,200},  % \upharpoonright
    "17  =  {200,200},  % \downharpoonright
    "18  =  {200,200},  % \upharpoonleft
    "19  =  {200,200},  % \downharpoonleft
    "1A  =  { 80,100},  % \rightarrowtail
    "1B  =  { 80,100},  % \leftarrowtail
    "1C  =  { 50, 50},  % \leftrightarrows
    "1D  =  { 50, 50},  % \rightleftarrows
    "1E  =  {250,   },  % \Lsh
    "1F  =  {   ,250},  % \Rsh
    "20  =  {100,100},  % \rightsquigarrow
    "21  =  {100,100},  % \leftrightsquigarrow
    "22  =  {100, 50},  % \looparrowleft
    "23  =  { 50,100},  % \looparrowright
    "24  =  { 50, 80},  % \circeq
    "25  =  {   ,100},  % \succsim
    "26  =  {   ,100},  % \gtrsim
    "27  =  {   ,100},  % \gtrapprox
    "28  =  {150, 50},  % \multimap
    "2B  =  {100,150},  % \doteqdot
    "2C  =  {100,150},  % \triangleq
    "2D  =  {100, 50},  % \precsim
    "2E  =  {100, 50},  % \lesssim
    "2F  =  { 50, 50},  % \lessapprox
    "30  =  {100, 50},  % \eqslantless
    "31  =  { 50, 50},  % \eqslantgtr
    "32  =  {100, 50},  % \curlyeqprec
    "33  =  { 50,100},  % \curlyeqsucc
    "34  =  {100, 50},  % \preccurlyeq
    "36  =  { 50,   },  % \leqslant
    "38  =  {   , 50},  % \backprime
    "39  =  {250,250},  % \dabar@ : the dash bar in \dash(left,right)arrow
    "3C  =  { 50,100},  % \succcurlyeq
    "3E  =  {   , 50},  % \geqslant
    "40  =  {   , 50},  % \sqsubset
    "41  =  { 50,   },  % \sqsupset
    "42  =  {   ,150},  % \vartriangleright, \rhd
    "43  =  {150,   },  % \vartriangleleft, \lhd
    "44  =  {   ,100},  % \trianglerighteq, \unrhd
    "45  =  {100,   },  % \trianglelefteq, \unlhd
    "46  =  {100,100},  % \bigstar
    "48  =  { 50, 50},  % \blacktriangledown
    "49  =  {   ,100},  % \blacktriangleright
    "4A  =  {100,   },  % \blacktriangleleft
    "4B  =  {   ,150},  % \dashrightarrow (the arrow)
    "4C  =  {150,   },  % \dashleftarrow
    "4D  =  { 50, 50},  % \vartriangle
    "4E  =  { 50, 50},  % \blacktriangle
    "4F  =  { 50, 50},  % \triangledown
    "50  =  { 50, 50},  % \eqcirc
    "56  =  {   ,150},  % \Rrightarrow
    "57  =  {150,   },  % \Lleftarrow
    "58  =  {100,300},  % \checkmark
    "5C  =  { 50, 50},  % \angle
    "5D  =  { 50, 50},  % \measuredangle
    "5E  =  { 50, 50},  % \sphericalangle
```

```
     "5F  =  {   , 50},  % \varpropto
     "60  =  {100,100},  % \smallsmile
     "61  =  {100,100},  % \smallfrown
     "62  =  { 50,   },  % \Subset
     "63  =  {   , 50},  % \Supset
     "66  =  {150,150},  % \curlywedge
     "67  =  {150,150},  % \curlyvee
     "68  =  { 50,150},  % \leftthreetimes
     "69  =  {100, 50},  % \rightthreetimes
     "6C  =  { 50, 50},  % \bumpeq
     "6D  =  { 50, 50},  % \Bumpeq
     "6E  =  {100,   },  % \lll
     "6F  =  {   ,100},  % \ggg
     "70  =  { 50,100},  % \ulcorner
     "71  =  {100, 50},  % \urcorner
     "75  =  {150,200},  % \dotplus
     "76  =  { 50,100},  % \backsim
     "78  =  { 50,100},  % \llcorner
     "79  =  {100, 50},  % \lrcorner
     "7C  =  {100,100},  % \intercal
     "7D  =  { 50, 50},  % \circledcirc
     "7E  =  { 50, 50},  % \circledast
     "7F  =  { 50, 50}   % \circleddash
```

Remaining slots in the source file.

```
   }

⟨/msa⟩
```

Symbol font 'b'.

```
⟨*msb⟩
\SetProtrusion
   [ name     = AMS-b ]
   { encoding = U,
     family   = msb  }
   {
       A  =  { 50, 50},  % \mathbb
       C  =  { 50, 50},
       G  =  {   , 50},
       L  =  {   , 50},
       P  =  {   , 50},
       R  =  {   , 50},
       T  =  {   , 50},
       V  =  { 50, 50},
       X  =  { 50, 50},
       Y  =  { 50, 50},
     "00  =  { 50, 50},  % \lvertneqq
     "01  =  { 50, 50},  % \gvertneqq
     "02  =  { 50, 50},  % \nleq
     "03  =  { 50, 50},  % \ngeq
     "04  =  {100, 50},  % \nless
     "05  =  { 50,150},  % \ngtr
     "06  =  {100, 50},  % \nprec
     "07  =  { 50,150},  % \nsucc
     "08  =  { 50, 50},  % \lneqq
     "09  =  { 50, 50},  % \gneqq
     "0A  =  {100,100},  % \nleqslant
     "0B  =  {100,100},  % \ngeqslant
     "0C  =  {100, 50},  % \lneq
     "0D  =  { 50,100},  % \gneq
     "0E  =  {100, 50},  % \npreceq
     "0F  =  { 50,100},  % \nsucceq
     "10  =  { 50,   },  % \precnsim
     "11  =  { 50, 50},  % \succnsim
     "12  =  { 50, 50},  % \lnsim
```

```
    "13  =  { 50, 50},  % \gnsim
    "14  =  { 50, 50},  % \nleqq
    "15  =  { 50, 50},  % \ngeqq
    "16  =  { 50, 50},  % \precneqq
    "17  =  { 50, 50},  % \succneqq
    "18  =  { 50, 50},  % \precnapprox
    "19  =  { 50, 50},  % \succnapprox
    "1A  =  { 50, 50},  % \lnapprox
    "1B  =  { 50, 50},  % \gnapprox
    "1C  =  {150,200},  % \nsim
    "1D  =  { 50, 50},  % \ncong
    "1E  =  {100,150},  % \diagup
    "1F  =  {100,150},  % \diagdown
    "20  =  {100, 50},  % \varsubsetneq
    "21  =  { 50,100},  % \varsupsetneq
    "22  =  {100, 50},  % \nsubseteqq
    "23  =  { 50,100},  % \nsupseteqq
    "24  =  {100, 50},  % \subsetneqq
    "25  =  { 50,100},  % \supsetneqq
    "26  =  {100, 50},  % \varsubsetneqq
    "27  =  { 50,100},  % \varsupsetneqq
    "28  =  {100, 50},  % \subsetneq
    "29  =  { 50,100},  % \supsetneq
    "2A  =  {100, 50},  % \nsubseteq
    "2B  =  { 50,100},  % \nsupseteq
    "2C  =  { 50,100},  % \nparallel
    "2D  =  {100,150},  % \nmid
    "2E  =  {150,150},  % \nshortmid
    "2F  =  {100,100},  % \nshortparallel
    "30  =  {   ,150},  % \nvdash
    "31  =  {   ,150},  % \nVdash
    "32  =  {   ,100},  % \nvDash
    "33  =  {   ,100},  % \nVDash
    "34  =  {   ,100},  % \ntrianglerighteq
    "35  =  {100,   },  % \ntrianglelefteq
    "36  =  {100,   },  % \ntriangleleft
    "37  =  {   ,100},  % \ntriangleright
    "38  =  {100,200},  % \nleftarrow
    "39  =  {100,200},  % \nrightarrow
    "3A  =  {100,100},  % \nLeftarrow
    "3B  =  { 50,100},  % \nRightarrow
    "3C  =  {100,100},  % \nLeftrightarrow
    "3D  =  {100,200},  % \nleftrightarrow
    "3E  =  { 50, 50},  % \divideontimes
    "3F  =  { 50, 50},  % \varnothing
    "60  =  {200,   },  % \Finv
    "61  =  {   , 50},  % \Game
    "68  =  {100,100},  % \eqsim
    "69  =  { 50,   },  % \beth
    "6A  =  { 50,   },  % \gimel
    "6B  =  {150,   },  % \daleth
    "6C  =  {200,   },  % \lessdot
    "6D  =  {   ,200},  % \gtrdot
    "6E  =  {100,200},  % \ltimes
    "6F  =  {150,100},  % \rtimes
    "70  =  { 50,100},  % \shortmid
    "71  =  { 50, 50},  % \shortparallel
    "72  =  {200,300},  % \smallsetminus
    "73  =  {100,200},  % \thicksim
    "74  =  { 50,100},  % \thickapprox
    "75  =  { 50, 50},  % \approxeq
    "76  =  { 50,100},  % \succapprox
    "77  =  { 50, 50},  % \precapprox
    "78  =  {100,100},  % \curvearrowleft
    "79  =  { 50,150},  % \curvearrowright
    "7A  =  { 50,200},  % \digamma
    "7B  =  {100, 50},  % \varkappa
    "7F  =  {200,   }   % \backepsilon
```

Remaining slots in the source file.

```
  }

⟨/msb⟩
```

2.8.8 Euler

Euler Roman font (package euler).

```
⟨*eur⟩
\SetProtrusion
  [ name     = euler ]
  { encoding = U,
    family   = eur  }
  {
    "01  =  {100,100},
    "03  =  {100,150},
    "06  =  {   ,100},
    "07  =  {100,150},
    "08  =  {100,100},
    "0A  =  {100,100},
    "0B  =  {   , 50},
    "0C  =  {   ,100},
    "0D  =  {100,100},
    "0E  =  {   ,100},
    "0F  =  {100,100},
    "10  =  {100,100},
    "13  =  {   ,100},
    "14  =  {   ,100},
    "15  =  {   , 50},
    "16  =  {   , 50},
    "17  =  { 50,100},
    "18  =  { 50,100},
    "1A  =  {   , 50},
    "1B  =  {   , 50},
    "1C  =  { 50,100},
    "1D  =  { 50,100},
    "1E  =  { 50,100},
    "1F  =  { 50,100},
    "20  =  {   , 50},
    "21  =  {   , 50},
    "22  =  { 50,100},
    "24  =  {   , 50},
    "27  =  { 50,100},
      1  =  {100,100},
      7  =  { 50,100},
    "3A  =  {300,500},
    "3B  =  {200,400},
    "3C  =  {200,100},
    "3D  =  {200,200},
    "3E  =  {100,200},
      A  =  {   ,100},
      D  =  {   , 50},
      J  =  { 50,   },
      K  =  {   , 50},
      L  =  {   , 50},
      Q  =  {   , 50},
      T  =  { 50,   },
      X  =  { 50, 50},
      Y  =  { 50,   },
      h  =  {   , 50},
       k  =  {   , 50}
   }

```

Extended by the `eulervm` package.

```
\SetProtrusion
   [ name     = euler-vm,
     load     = euler ]
   { encoding = U,
     family   = zeur  }
   {
     "28  =  {100,200},
     "29  =  {100,200},
     "2A  =  {100,150},
     "2B  =  {100,150},
     "2C  =  {200,300},
     "2D  =  {200,300},
     "2E  =  {   ,100},
     "2F  =  {100,   },
     "3F  =  {150,150},
     "5B  =  {   ,100},
     "5E  =  {100,100},
     "5F  =  {100,100},
     "80  =  {   , 50},
     "81  =  {200,250},
     "82  =  {100,200}
   }

⟨/eur⟩
```

Euler Script font (`eucal`).

```
⟨*eus⟩
\SetProtrusion
   [ name     = euscript ]
   { encoding = U,
     family   = eus  }
   {
       A  =  {100,100},
       B  =  { 50,100},
       C  =  { 50, 50},
       D  =  { 50,100},
       E  =  { 50,100},
       F  =  { 50,   },
       G  =  { 50,   },
       H  =  {   ,100},
       K  =  {   , 50},
       L  =  {   ,150},
       M  =  {   , 50},
       N  =  {   , 50},
       O  =  { 50, 50},
       P  =  { 50, 50},
       T  =  {   ,100},
       U  =  {   , 50},
       V  =  { 50, 50},
       W  =  { 50, 50},
       X  =  { 50, 50},
       Y  =  { 50,   },
       Z  =  { 50,100},
     "00  =  {250,250},
     "18  =  {200,200},
     "3A  =  {200,150},
     "40  =  {   ,100},
     "5E  =  {100,100},
     "5F  =  {100,100},
     "66  =  { 50,   },
     "67  =  {   , 50},
     "6E  =  {200,200}
   }

\SetProtrusion
   [ name     = euscript-vm,
     load     = euscript ]
   { encoding = U,
     family   = zeus  }
   {
     "01  =  {600,600},
     "02  =  {200,200},
     "03  =  {200,200},
     "04  =  {200,200},
     "05  =  {150,150},
     "06  =  {200,200},
     "07  =  {200,200},
     "08  =  {100,100},
     "09  =  {100,100},
     "0A  =  {100,100},
     "0B  =  {100,100},
     "0C  =  {100,100},
     "0D  =  {100,100},
     "0E  =  {150,150},
     "0F  =  {100,100},
     "10  =  {150,150},
     "11  =  {100,100},
     "12  =  {150,100},
     "13  =  {100,150},
     "14  =  {150,100},
     "15  =  {100,150},
     "16  =  {200,100},
     "17  =  {100,200},
     "19  =  {150,150},
     "1A  =  {150,100},
     "1B  =  {100,150},
     "1C  =  {100,100},
     "1D  =  {100,100},
     "1E  =  {250,100},
     "1F  =  {100,250},
     "20  =  {150,200},
     "21  =  {150,200},
     "22  =  {150,150},
     "23  =  {150,150},
     "24  =  {100,200},
     "25  =  {150,150},
     "26  =  {150,150},
     "27  =  {100,100},
     "28  =  {100,100},
     "29  =  {100,150},
     "2A  =  {100,100},
     "2B  =  {100,100},
     "2C  =  {100,100},
     "2D  =  {150,150},
     "2E  =  {150,150},
     "2F  =  {100,100},
     "30  =  {100,100},
     "31  =  {100,100},
     "32  =  {100,100},
     "33  =  {100,100},
     "34  =  {100,100},
     "35  =  {100,100},
     "3E  =  {150,150},
     "3F  =  {150,150},
     "60  =  {   ,200},
     "61  =  {200,   },
     "62  =  {100,100},
     "63  =  {100,100},
     "64  =  {100,100},
     "65  =  {100,100},
     "68  =  {300,   },
     "69  =  {   ,300},
     "6C  =  {100,100},
     "6D  =  {100,100},
     "6F  =  {100,100},
     "72  =  {100,100},
     "73  =  {200,100},
     "76  =  {   ,100},
     "77  =  {100,   },
     "78  =  { 50, 50},
     "79  =  {100,100},
     "7A  =  {100,100},
     "7D  =  {150,150},
     "7E  =  {100,100},
     "A8  =  {100,100},
     "A9  =  {100,100},
     "AB  =  {200,200},
     "BA  =  {   ,200},
     "BB  =  {   ,200},
     "BD  =  {200,200},
     "DE  =  {200,200}
   }

⟨/eus⟩
```

Euler Fraktur font (eufrak).

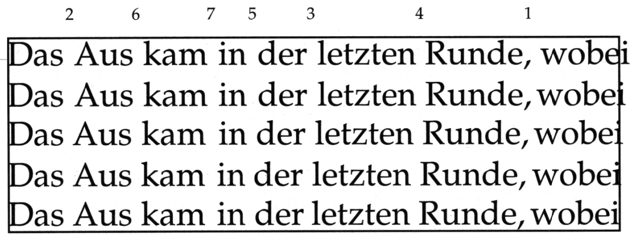
```
⟨*euf⟩
\SetProtrusion
   [ name      = mathfrak ]
   { encoding = U,
     family   = euf  }
   {
       A  =  {   , 50},
       B  =  {   , 50},
       C  =  { 50, 50},
       D  =  {   , 80},
       E  =  { 50,   },
       G  =  {   , 50},
       L  =  {   , 80},
       O  =  {   , 50},
       T  =  {   , 80},
       X  =  { 80, 50},
       Z  =  { 80, 50},
       b  =  {   , 50},
       c  =  {   , 50},
       k  =  {   , 50},
       p  =  {   , 50},
       q  =  { 50,   },
       v  =  {   , 50},
       w  =  {   , 50},
       x  =  {   , 50},
       1  =  {100,100},
       2  =  { 80, 80},
       3  =  { 80, 50},
       4  =  { 80, 50},
       7  =  { 50, 50},
     "12  =  {500,500},
     "13  =  {500,500},
       !  =  {   ,200},
       '  =  {200,300},
       (  = {200,   },
       )  = {   ,200},
       *  = {200,200},
       +  = {200,250},
       -  = {200,200},
      {,} = {300,300},
       .  = {400,400},
      {=} = {200,200},
       :  = {   ,200},
       ;  = {   ,200},
       ]  = {   ,200}
   }

⟨/euf⟩
⟨/cfg-u⟩
```

2.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages `eurosans`, `europs`), ITC Euro fonts (package `euroitc`) and `marvosym`[17]). The `euroitc` settings are hidden in the package itself (1.3.8) for 'free software' compliance reasons. (Not quite sure whether this is what Karl really had in mind . . .)

```
⟨*cfg-e⟩
\SetProtrusion
⟨zpeu⟩   { encoding = U,
⟨mvs⟩   { encoding = {OT1,U},
⟨zpeu⟩     family   = zpeu }
⟨mvs⟩     family   = mvs }
   {
⟨zpeu⟩     E = {50, }
⟨mvs⟩     164 = {50,50},   % \EUR
⟨mvs⟩     068 = {50,-100}  % \EURdig
   }

⟨*zpeu⟩
\SetProtrusion
   { encoding = U,
     family   = zpeu,
     shape    = it* }
   {
     E = {100,-50}
   }

\SetProtrusion
   { encoding = U,
     family   = {zpeus,eurosans} }
   {
     E = {100,50}
   }

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

⟨/zpeu⟩
⟨/cfg-e⟩
```

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



Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei
Das Aus kam in der letzten Runde, wobei

Figure 1:

Example of interword spacing (from: M. Siemoneit, *Typographisches Gestalten*, Frankfurt/M. 1989).
The numbers indicate the preference for shrinking the interword space.

2.9 Interword spacing

Default unit is space.

```
⟨*m-t|cmr⟩
%%% ----------------------------------------------------------------------
%%% INTERWORD SPACING

⟨/m-t|cmr⟩
⟨*m-t⟩
\SetExtraSpacing
   [ name = default ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
   {
```

These settings are only a first approximation. The following reasoning is from a mail from *Ulrich Dirr*, who also provided the sample in figure 1. I do not claim to have coped with the task.

> ‘The idea is – analog to the tables for expansion and protrusion – to have tables for optical reduction/expansion of spaces in dependence of the actual character so that the distance between words is optically equal.
>
> When reducing distances the (weighting) order is:

- after commas

```
         {,} = { ,-500,500},
```

- in front of capitals which have optical more room on their left side, e.g., ‘A’, ‘J’, ‘T’, ‘V’, ‘W’, and ‘Y’ [this is not yet possible – RS]
- in front of capitals which have circle/oval shapes on their left side, e.g., ‘C’, ‘G’, ‘O’, and ‘Q’ [ditto – RS]
- after ‘r’ (because of the bigger optical room on the righthand side)

```
          r  = { ,-300,300},
```

- [before or] after lowercase characters with ascenders

```
          b  = { ,-200,200},
          d  = { ,-200,200},
          f  = { ,-200,200},
          h  = { ,-200,200},
          k  = { ,-200,200},
          l  = { ,-200,200},
          t  = { ,-200,200},
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e.g., ‘v’, or ‘w’

```
          c  = { ,-100,100},
          p  = { ,-100,100},
          v  = { ,-100,100},
          w  = { ,-100,100},
          z  = { ,-100,100},
          x  = { ,-100,100},
          y  = { ,-100,100},
```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```
          i  = { , 50, -50},
          m  = { , 50, -50},
          n  = { , 50, -50},
          u  = { , 50, -50},
```

- after colon and semicolon

```
          :  = { ,200,-200},
          ;  = { ,200,-200},
```

- after punctuation which ends a sentence, e.g., period, exclamation mark, question mark

```
          .  = { ,250,-250},
          !  = { ,250,-250},
          ?  = { ,250,-250}
```

The order has to be reversed when enlarging is needed.’

```
   }

⟨/m-t⟩
```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a `factor` < 1000.)
- Should the first parameter also be used? (Probably.)
- What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTEX bug with spacing in combination with a non-zero `\spaceskip` (reported by *Axel Berger*):

```
\parfillskip0pt
\rightskip0pt plus 1em
\spaceskip\fontdimen2\font
 test test\par
\pdfadjustinterwordglue2
\stbscode\font`t=-50
 test test
\bye
```

Some more characters in T2A.[18]

```
⟨*cmr⟩
\SetExtraSpacing
   [ name     = T2A,
     load     = default ]
   { encoding = T2A,
     family   = cmr }
   {
      \cyrg  = { ,-300,300},
      \cyrb  = { ,-200,200},
      \cyrk  = { ,-200,200},
      \cyrs  = { ,-100,100},
      \cyrr  = { ,-100,100},
```

[18] Contributed by *Karl Karlsson*.

```
      \cyrh  = { ,-100,100},
      \cyru  = { ,-100,100},
      \cyrt  = { , 50, -50},
      \cyrp  = { , 50, -50},
      \cyri  = { , 50, -50},
      \cyrishrt  = { , 50, -50},
   }

```

2.9.1 Nonfrenchspacing

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

From the TeXbook:

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

The 'extra space' (`\fontdimen 7`) for Computer Modern Roman is a third of `\fontdimen 2`, i.e., 333.

```
\SetExtraSpacing
   [ name     = nonfrench-cmr,
     load     = default,
     context  = nonfrench ]
   { encoding = {OT1,T1,LY1,OT4,QX,T5},
     family   = cmr }
   {
```

`latex.ltx` has:

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

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

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

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

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

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

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

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

```
}
```

```
   }

⟨/cmr⟩
```

fontinst, however, which is also used to create the psnfss font metrics, sets \fontdimen 7 to 240 by default. Therefore, the fallback settings use this value for the first component.

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

```

Empty settings to prevent spurious warnings.

```
\SetExtraSpacing
  [ name = empty ]
  { encoding = {TS1} }
  { }

```

2.10 Additional kerning

Default unit is 1 em.

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

```

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

```
\SetExtraKerning
  [ name = empty ]
  { encoding = {OT1,T1,T2A,LY1,OT4,QX,T5,TS1} }
  { }

```

2.10.1 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[19] claims it should be a quarter of an em, which seems too much to me; then again, it also says that this *was* a thin space in French typography.)

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

[19] https://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

```
     ;  = {500, }, % ~ \thinspace
     !  = {500, },
     ?  = {500, }
   }

```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfTEX.

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

\SetExtraKerning
   [ name     = french-guillemets-OT1,
     context  = french-guillemets,
     load     = french-default,
     unit     = space   ]
   { encoding = OT1     }
   { }

```

2.10.2 Turkish

```
\SetExtraKerning
   [ name     = turkish,
     context  = turkish ]
   { encoding = {OT1,T1,LY1} }
   {
     :  = {167, }, % = \thinspace
     !  = {167, },
    {=} = {167, }
   }

⟨/m-t⟩
⟨/config⟩
```

3 OpenType configuration files

These are the configuration files for the following OpenType fonts:[20]

- Latin Modern Roman
- New Computer Modern[21]
- Charis SIL
- EB Garamond
- Palatino[22]

The settings are typeset in the respective font.

3.1 Character inheritance

OpenType fonts may differ considerably in how complete their arsenal of glyphs is. Therefore, each font family should have their own inheritance settings.

```

%%% -----------------------------------------------------------------------
%%% INHERITANCE

%% for xetex (EU1) and luatex (EU2), resp. both (TU)
```

3.1.1 Latin Modern Roman/New Computer Modern

```
⟨*LatinModernRoman|NewComputerModern⟩
\DeclareCharacterInheritance
   { encoding = {TU,EU1,EU2},
⟨LatinModernRoman⟩      family   = Latin Modern Roman }
⟨NewComputerModern⟩      family   = {New Computer Modern} }
  {
    A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǻ,Ȁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ặ,
⟨LatinModernRoman⟩          A  % Greek
⟨NewComputerModern⟩          A,A,Ạ,A,Ă,Ā  % ῾A,῍A,῝A,῎A,῞A,῏A,῟A,᾿Ạ,῾Ạ,῍Ạ,῝Ạ,῎Ạ,῞Ạ,῏Ạ,῟Ạ,`A,  % Greek
        },
    Æ = {Ǽ},
    B = {Ƀ,
        B},  % Greek
    C = {Ç,Ć,Ĉ,Ċ,Č},
    D = {Đ,Ď,Ð,Ḍ,Ḏ},
    E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ẹ,Ẻ,Ẽ,Ề,Ế,Ễ,Ể,Ệ,
        E},  % Greek
⟨NewComputerModern⟩  (l)E = {E,᾿E,῾E,῍E,῝E,῎E,῞E,`E,E}, % Greek accents fully protruded left
    G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ},
    H = {Ĥ,Ħ,Ḥ,Ḧ,Ḫ,
⟨LatinModernRoman⟩          H  % Greek
⟨NewComputerModern⟩          H,Ḥ  % Greek
        },
⟨NewComputerModern⟩    (l)H = {᾿H,῾H,῍H,῝H,῎H,῞H,῏H,῟H,᾿Ḥ,῾Ḥ,῍Ḥ,῝Ḥ,῎Ḥ,῞Ḥ,῏Ḥ,῟Ḥ}, % Greek accents fully protruded left
⟨NewComputerModern⟩  %(l)/uni1FCC.alt = {/uni1F98.alt},
    I = {Ì,Í,Î,Ï,Ĩ,Ī,Ĭ,Į,İ,Ȉ,Ỉ,Ị,
⟨LatinModernRoman⟩         I  % Greek
⟨NewComputerModern⟩         I,Ĭ,Ī  % Greek
```

[20] This is file `microtype-utf.dtx`.
[21] These settings have been contributed by *Antonis Tsolomitis*.
[22] These settings have been contributed by *Loren B. Davis*.

```
        },
⟨NewComputerModern⟩  (l)I = {Ἰ,Ἱ,Ἲ,Ἳ,Ἴ,Ἵ,Ἶ,Ἷ,Ὶ,Ι}, % Greek
    J = {Ĵ},
    K = {Ķ,
        Κ},  % Greek
    L = {Ĺ,Ļ,Ł,Ḷ}, % Ľ,Ŀ,Ḹ
    M = {Μ},  % Greek
    N = {Ñ,Ń,Ņ,Ň,Ṅ,Ṇ,
        Ν},  % Greek
    O = {Ò,Ó,Ô,Õ,Ö,Ø,Ō,Ŏ,Ő,Ọ,Ơ,Ǫ,Ǿ,Ȍ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ỡ,Ở,Ợ,
        Ο},  % Greek
⟨NewComputerModern⟩  (l)O = {Ο,Ὀ,Ὁ,Ὂ,Ὃ,Ὄ,Ὅ,Ο}, % Greek accents except O that has indep. protrusion numbers (below)
    P = {Ρ},  % Greek
⟨NewComputerModern⟩  (l)P = {Ῥ}, % Greek accents fully protruded left
    R = {Ŕ,Ŗ,Ř,Ȑ,Ṛ,Ṙ,Ṝ},
    S = {Ś,Ŝ,Ş,Š,Ș,Ṣ},
    T = {Ţ,Ť,Ț,Ṭ,Ṯ,
        Τ},  % Greek
    U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ư,Ȕ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
    W = {Ŵ,Ẁ,Ẃ,Ẅ},
    X = {Χ},  % Greek
    Y = {Ý,Ŷ,Ÿ,Ỵ,Ỷ,Ỹ},
⟨NewComputerModern⟩     Υ = {Ϋ,Ῠ,Ῡ},
⟨NewComputerModern⟩  (l)Υ = {Υ,Ὺ,Υ,Ὑ,Ὓ,Ὕ,Ὗ},
    Z = {Ź,Ż,Ž,Ẓ,
        Ζ},  % Greek
    a = {à,á,â,ã,ä,å,ā,ă,ą,ǻ,ȁ,ạ,ả,ấ,ầ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ},
    æ = {ǽ},
    c = {ç,ć,ĉ,ċ,č},
    d = {đ,ḍ,ḏ},
    e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ẹ,ẻ,ẽ,ế,ề,ể,ễ,ệ},
     f = {ﬀ}, % Unicode 64256, glyph name in Latin Modern Roman: /f_f ; in New Computer Modern: /ff
    g = {ĝ,ğ,ġ,ģ,ǧ,ǵ},
    h = {ĥ,ħ,ḥ,ḧ,ḫ},
    i = {ì,í,î,ï,ĩ,ī,ĭ,į,ı,ȉ,ỉ,ị},
    j = {ĵ},
    k = {ķ},
    l = {ĺ,ļ,ł,ḷ,ḹ}, % ľ,ŀ
    n = {ñ,ń,ņ,ň,ṅ,ṇ},
    o = {ò,ó,ô,õ,ö,ø,ō,ŏ,ő,ọ,ơ,ǫ,ǿ,ȍ,ỏ,ố,ồ,ổ,ỗ,ộ,ớ,ờ,ỡ,ở,ợ
⟨NewComputerModern⟩        ,ο,ό,ὀ,ὁ,ὂ,ὃ,ὄ,ὅ,ὸ,ό % Greek
        },
    r = {ŕ,ŗ,ř,ȑ,ṛ,ṙ,ṝ},
    s = {ś,ŝ,ş,š,ș,ṣ},
    t = {ţ,ț,ṭ,ṯ,ẗ}, % ť
    u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ư,ȕ,ụ,ủ,ứ,ừ,ử,ữ,ự},
    w = {ŵ,ẁ,ẃ,ẅ},
    y = {ý,ŷ,ÿ,ỳ,ỵ,ỷ,ỹ},
    z = {ź,ż,ž,ẓ},
⟨*NewComputerModern⟩
    α = {ἁ,ἂ,ἃ,ἄ,ἅ,ἆ,ἇ,ᾀ,ᾁ,ᾂ,ᾃ,ᾄ,ᾅ,ᾆ,ᾇ,ὰ},
    ε = {έ,ἐ,ἑ,ἒ,ἓ,ἔ,ἕ,ὲ,έ},
    η = {ἠ,ἡ,ἢ,ἣ,ἤ,ἥ,ἦ,ἧ,ᾐ,ᾑ,ᾒ,ᾓ,ᾔ,ᾕ,ᾖ},
    ι = {ἰ,ἱ,ὶ,ί,ῐ,ῑ,ϊ,ί},
    ϊ = {ἲ,ἳ,ἴ,ἵ,ἶ,ἷ,ῐ,ῑ,ί},
    υ = {ύ,ὺ,ύ,ὑ,ὓ,ὕ,ὗ,ϋ,ῠ,ῡ,ϋ,ΰ},
    ω = {ώ,ὠ,ὡ,ὢ,ὣ,ὤ,ὥ,ὦ,ὧ,ᾠ,ᾡ,ᾢ,ᾣ,ᾤ,ᾥ,ᾦ},
⟨/NewComputerModern⟩
  }
⟨/LatinModernRoman|NewComputerModern⟩
```

3.1.2 Charis SIL

```
⟨*CharisSIL⟩
\DeclareCharacterInheritance
   { encoding = {TU,EU1,EU2},
     family   = Charis SIL }

  { A  =  {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ȧ,Ǻ,Ȁ,Ȃ,Ḁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ꜳ,
           А,Ӑ,Ӓ},  %  Cyrillic
    Æ  =  {Ǣ,
           Ǽ,Ӕ},  %  Cyrillic
    B  =  {Ḃ,Ḅ,Ḇ,
           В},  %  Cyr
    C  =  {Ç,Ć,Ĉ,Ḉ,Ċ,Č,
           С,Ҫ},  %  Cyr
    D  =  {Ď,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ,Đ},
    E  =  {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ȩ,Ȇ,Ḕ,Ḗ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,Ể,Ễ,Ệ,
           Е,Ѐ,Ё,Ӗ},  %  Cyr
    F  =  {Ḟ},
    G  =  {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ,Ḡ,Ԍ},
    H  =  {Ĥ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ,
           Н,Ң,Ԋ,Ӈ,Ӊ},  %  Cyr
    I  =  {Ì,Í,Î,Ȉ,Ï,Į,Ĩ,Ị,Ī,İ,Ǐ,Ỉ,Ȋ,Ĭ,Ḭ,Ï,
           І,Ї,Ӏ,ӏ},  %  Cyr
    J  =  {Ĵ,
           Ј},  %  Cyr
    K  =  {Ķ,Ǩ,Ḱ,Ḳ,Ḵ,
           К,Ќ,Қ,Ҝ,Ҟ,Ҡ,Ӄ,Ԟ},  %  Cyr
    L  =  {Ĺ,Ļ,Ľ,Ḷ,Ḹ,Ḻ,Ḽ},  %  L·
    M  =  {Ḿ,Ṁ,Ṃ,
           М,Ӎ},  %  Cyr
    N  =  {Ñ,Ń,Ņ,Ň,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ,
           И,Й,Ѝ,Ҋ,Ӣ,Ӥ},  %  Cyr
    O  =  {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Ǒ,Ǫ,Ǭ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ṍ,Ṏ,Ṑ,Ṓ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ,
           О,Ө,Ӧ,Ѳ,Ӫ,  %  Cyr
           Θ},  %  Greek
    P  =  {Ṕ,Ṗ,
           Р,Ҏ},    %  Cyr
    Q  =  {Ԛ},  %  Cyr
    R  =  {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ},
    S  =  {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ,
           Ѕ},  %  Cyr
    T  =  {Ţ,Ť,Ț,Ṫ,Ṭ,Ṯ,Ṱ,
           Т,Ҭ},  %  Cyr
    U  =  {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
    V  =  {Ṽ,Ṿ},
    W  =  {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ,
           Ԝ},  %  Cyr
    X  =  {Ẋ,Ẍ,
           Х,Ҳ,Ӽ,Ӿ},    %  Cyr
    Y  =  {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ,
           Ү,Ұ},  %  Cyr
    Z  =  {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ},
    a  =  {à,á,â,ã,ä,å,ā,ă,ą,ǎ,ǟ,ǡ,ȧ,ǻ,ȁ,ȃ,ḁ,ạ,ả,ầ,ấ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ,ặ,  %  aᵓ
           а,ӑ,ӓ},  %  Cyr
    æ  =  {ǽ,
           ӕ},  %  Cyr
    b  =  {ḃ,ḅ,ḇ},
    c  =  {ç,ć,ĉ,ċ,č,ḉ,
           с,ҫ},  %  Cyr
    d  =  {ď,ḋ,ḍ,ḏ,ḑ,ḓ},
    e  =  {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ȩ,ȇ,ḕ,ḗ,ḙ,ḛ,ḝ,ẹ,ẻ,ẽ,ề,ế,ể,ễ,ệ,
           е,ѐ,ё,ӗ},  %  Cyr
    f  =  {ḟ,ﬀ},  %  /f_f
```

```
      g  =  {ĝ,ğ,ġ,ģ,ǧ,ǵ,ḡ},
      h  =  {ĥ,ȟ,ħ,ḥ,ḧ,ḩ,ḫ,ẖ,
             ћ,һ},  %  Cyr
      i  =  {ı,ì,í,î,ȉ,ǐ,ï,į,ī,ị,ĩ,ḭ,ḯ,ỉ,ȋ,ĭ,
             і,ї},  %  Cyr
      j  =  {ĵ,ǰ,
             ј},  %  Cyr
      k  =  {ķ,ǩ,ḱ,ḳ,ḵ},
      l  =  {ĺ,ļ,ḻ,ḷ,ḹ,ḽ},  %  ľ,ŀ
      m  =  {ḿ,ṁ,ṃ},
      n  =  {ñ,ń,ņ,ň,ǹ,ṅ,ṇ,ṉ,ṋ},  %  ŉ
      o  =  {ò,ó,ô,õ,ö,ō,ŏ,ő,ǒ,ǫ,ǭ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ṍ,ṏ,ṑ,ṓ,ọ,ỏ,ồ,ố,ổ,ỗ,ộ,ớ,ờ,ở,ỡ,ợ,
             о,ѳ,ӧ,ө,ӫ},  %  Cyr
      p  =  {ṕ,ṗ,
             р,ҏ},  %  Cyr
      q  =  {ԛ},  %  Cyr
      r  =  {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
      s  =  {ś,ŝ,ş,š,ș,ṡ,ṣ,ṥ,ṧ,ṩ,
             ѕ},  %  Cyr
      t  =  {ţ,ț,ṫ,ṭ,ṯ,ṱ,ẗ},  %  ť
      u  =  {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự},
      v  =  {ṽ,ṿ},
      w  =  {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ,
             ԝ},   %  Cyr
      x  =  {ẋ,ẍ,
             х,ҳ},  %  Cyr
      y  =  {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,ỵ,ỷ,ỹ,
             у,ў,ӯ,ӱ,ӳ},  %  Cyr
      z  =  {ź,ż,ž,ẑ,ẓ,ẕ},
    %  Cyrillic
      Г  =  {Ѓ,Ґ,Ғ,Ӷ,Ӻ},
      Ж  =  {Җ,Ӝ,Ӂ},
      З  =  {Ӟ,Ҙ},
      Л  =  {Ӆ},
      П  =  {Ԥ},
      У  =  {Ў,Ӯ,Ӱ,Ӳ},
      Ч  =  {Ҷ,Ҹ,Ӌ,Ӵ},
      Ы  =  {Ӹ},
      Ә  =  {Ӛ},
      Ҽ  =  {Ҿ},
      г  =  {ѓ,ґ,ғ,ӷ,ӻ},
      ж  =  {җ,ӂ,ӝ},
      з  =  {ҙ,ӟ},
      и  =  {й,ѝ,ҋ,ӣ,ӥ},
      к  =  {ќ,қ,ҝ,ҟ,ҡ,ӄ,ԟ},
      л  =  {ӆ},
      м  =  {ӎ},
      н  =  {ң,ҥ,ӈ,ӊ},
      п  =  {ԥ},
      т  =  {ҭ},
      х  =  {ҳ,ӿ},
      ч  =  {ҷ,ҹ,ӌ,ӵ},
      ш  =  {щ},
      ы  =  {ӹ},
      э  =  {ӭ},
      ҽ  =  {ҿ},
      ә  =  {ӛ},
      ү  =  {ұ},
      Γ  =  {Γ},  %  Greek
      Π  =  {Π},  %  Greek
    %  missing:  tipa,  math,  symbols,  ...
    }
⟨/CharisSIL⟩
```

3.1.3 EB Garamond

```
⟨*EBGaramond⟩
\DeclareCharacterInheritance
   { encoding = {TU,EU1,EU2},
     family   = EBGaramond }

   {
     A = {À,Á,Â,Ã,Ä,Å,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ǻ,Ȁ,Ȃ,Ȧ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ḁ,/Adieresis.deu,
          A,Ӑ,Ӓ,                    % Cyrillic
          A,Ά,Ἀ,Ἁ,Ἂ,Ἃ,Ἄ,Ἅ,Ἆ,Ἇ,ᾈ,ᾉ,ᾊ,ᾋ,ᾌ,ᾍ,ᾎ,ᾏ,Ᾰ,Ᾱ,Ὰ,Ά,ᾼ}, % Greek
% (l)A    {Ά,Ἀ,Ἁ,Ἂ,Ἃ,Ἄ,Ἅ,Ἆ,Ἇ,ᾈ,ᾉ,ᾊ,ᾋ,ᾌ,ᾍ,ᾎ,ᾏ,Ᾰ,Ᾱ,Ὰ,Ά,ᾼ}, % (accents not protruded)
     B = {Ḃ,Ḅ,Ḇ,
          B,                        % Cyrillic
          B},                       % Greek
     C = {Ç,Ć,Ĉ,Ċ,Č,Ḉ,
          C,C,Ҫ,                    % Cyrillic
          C},                       % Roman numeral
     D = {Ď,Đ,Ɖ,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ,
          Đ,Ɗ,                      % Cyrillic
          D},                       % Roman numeral
     E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ḗ,Ȩ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,Ể,Ễ,Ệ,Ḕ,Ȇ,
          Ѐ,Ё,Ӗ,E,                  % Cyrillic
          E},                       % Greek
  (l)E = {Έ,Ἐ,Ἑ,Ἒ,Ἓ,Ἔ,Ἕ,Ὲ,Έ},    % Greek (accents protruded)
     F = {Ḟ},
     G = {Ĝ,Ğ,Ġ,Ģ,Ǥ,Ǧ,Ǵ,Ḡ},
     H = {Ĥ,Ħ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ,H̦,Ⱨ,
          H,Ң,Ӊ,Ӈ,  %  Ҥ          % Cyrillic
          H},                       % Greek
  (l)H = {Ή,Ἠ,Ἡ,Ἢ,Ἣ,Ἤ,Ἥ,Ἦ,Ἧ,Ὴ,Ή,ᾘ,ᾙ,ᾚ,ᾛ,ᾜ,ᾝ,ᾞ,ᾟ},
     I = {Ì,Í,Î,Ï,Ĩ,Ī,Ĭ,Į,İ,Ǐ,Ȉ,Ȋ,Ḭ,Ḯ,Ỉ,Ị,
          I,Ї,I,                    % Cyrillic
          I,                        % Greek
          I,II,III},                % Roman numeral
  (l)I = {Ί,Ϊ,Ἰ,Ἱ,Ἲ,Ἳ,Ἴ,Ἵ,Ἶ,Ἷ,Ῐ,Ῑ,Ὶ,Ί}, % Greek
     J = {Ĵ,
          J},                       % Cyrillic
     K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ,Ⱪ,
          K,K},                     % Greek
     L = {Ĺ,Ļ,Ľ,Ŀ,Ł,Ḷ,Ḹ,Ḻ,Ḽ,Ⱡ,Ł,
          L},                       % Roman numeral
     M = {Ḿ,Ṁ,Ṃ,
          M,Ӎ,                      % Cyrillic
          M,                        % Greek
          M},                       % Roman numeral
     N = {Ñ,Ń,Ņ,Ň,Ŋ,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ,
          N},                       % Greek
    O = {Ò,Ó,Ô,Õ,Ö,Ø,Ō,Ŏ,Ő,Ǒ,Ǫ,Ǭ,Ǿ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ṍ,Ṏ,Ṑ,Ṓ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ,Ơ,/Odieresis.deu,
          O,Ӧ,Ө,Ӫ,                  % Cyrillic
          O,Ό,Ὀ,Ὁ,Ὂ,Ὃ,Ὄ,Ὅ,Ὸ,Ό}, % Greek
% (l)O = {Ό,Ὀ,Ὁ,Ὂ,Ὃ,Ὄ,Ὅ,Ὸ,Ό}, % (accents not protruded)
     P = {Ṕ,Ṗ,Ᵽ,
          P,Ҏ,                      % Cyrillic
          P},                       % Greek
  (l)P = {Ῥ},                       % Greek
     Q = {Q},                       % Cyrillic
     R = {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ,Ɽ,},
     S = {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ,
          S},                       % Cyrillic
     T = {Ţ,Ť,Ŧ,Ț,Ṫ,Ṭ,Ṯ,Ṱ,
          T,Ҭ,                      % Cyrillic
          T},                       % Greek
     U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự}, % /Udieresis.deu ?
     V = {Ṽ,Ṿ,/U.LAT,
          V},                       % Roman numeral
```

```
W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ,
      Ԝ},                           % Cyrillic
X = {Ẋ,Ẍ,
      Х,Ҳ,Ӽ,Ӿ,                      % Cyrillic
      Χ,                            % Greek
      Ⅹ},                           % Roman numeral
Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ,
      Ү,Ұ},   % Cyrillic
Z = {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ,Ȥ,
      Ζ},                           % Greek
a = {à,á,â,ã,ä,å,ā,ă,ą,ǎ,ǟ,ǡ,ǻ,ȁ,ȃ,ȧ,ḁ,ạ,ả,ấ,ầ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ,ặ,ả,
      а,ӑ,ӓ},                       % Cyrillic
b = {ḃ,ḅ,ḇ},
c = {ç,ć,ĉ,ċ,č,ḉ,
      с,ҫ,                          % Cyrillic
      ⅽ},                           % Roman numeral
d = {ď,đ,ḋ,ḍ,ḏ,ḑ,ḓ,
      ⅾ},                           % Roman numeral
e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȩ,ḕ,ḗ,ḙ,ḛ,ḝ,ẹ,ẻ,ẽ,ế,ề,ể,ễ,ệ,ȅ,ȇ,
      е,ѐ,ё,ӗ},                     % Cyrillic
f = {ḟ,ﬀ,/f.long,/f.DEU,/f_f},
ﬂ = {ﬄ,/longs_l,/longs_longs_l,/f_l},
ﬁ = {ﬃ,/longs_i,/longs_longs_i,/f_i},
/f.short = {/f_f.short},
g = {ĝ,ğ,ġ,ģ,ḡ,ǥ,ǧ,ǵ},
h = {ĥ,ħ,ḣ,ḥ,ḧ,ḩ,ḫ,ẖ,ȟ,ⱨ,
      һ,ԧ},                         % Cyrillic
i = {ì,í,î,ï,ĩ,ī,ĭ,į,ı,ǐ,ȉ,ȋ,ḭ,ỉ,ị,ḯ,/i.TRK,
      і,ї,                          % Cyrillic
      ⅰ,ⅱ,ⅲ},                       % Roman numeral
j = {ĵ,ǰ,
      ј},                           % Cyrillic
k = {ķ,ǩ,ḱ,ḳ,ḵ,ⱪ},
l = {ĺ,ļ,ľ,ŀ,ł,ḷ,ḹ,ḻ,ḽ,ƚ,
      ӏ,                            % palochka
      ⅼ},                           % Roman numeral
m = {ḿ,ṁ,ṃ,
      ⅿ},                           % Roman numeral
n = {ñ,ń,ņ,ň,ŋ,ṅ,ṇ,ṉ,ṋ,ǹ}, % ŉ
o = {ò,ó,ô,õ,ö,ø,ō,ŏ,ő,ǒ,ǫ,ǭ,ǿ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ṍ,ṏ,ṑ,ṓ,ọ,ỏ,ố,ồ,ổ,ỗ,ộ,ớ,ờ,ở,ỡ,ợ,
      о,ӧ},                         % Cyrillic
p = {ṕ,ṗ,
      р,ҏ},                         % Cyrillic
q = {ԛ},                            % Cyrillic
r = {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
s = {ś,ŝ,ş,š,ș,ṡ,ṣ,ṥ,ṧ,ṩ,
      ѕ},                           % Cyrillic
t = {ţ,ť,ŧ,ț,ṫ,ṭ,ṯ,ṱ,ẗ},
u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự,/u.LATmedi},
v = {ṽ,ṿ,
      ⅴ},                           % Roman numeral
w = {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ,
      ԝ},                           % Cyrillic
x = {ẋ,ẍ,
      х,ҳ,                          % Cyrillic
      ⅹ},                           % Roman numeral
y = {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,ỵ,ỷ,ỹ,
      у,ӯ,ӱ,ӳ,ў},                   % Cyrillic
z = {ź,ż,ž,ȥ,ẑ,ẓ,ẕ},
Æ = {Ǣ,Ǽ,
      Ӕ},                           % Cyrillic
æ = {ǣ,ǽ,
      ӕ},                           % Cyrillic
DZ = {Ǆ},
Dz = {ǅ},
    dz = {dž},
  % Smallcaps
    /a.sc = {/A.sc},
    /ae.sc = {/AE.sc},
    /d.sc = {/D.sc},
    /f.sc = {/F.sc},
    /g.sc = {/G.sc},
    /j.sc = {/J.sc},
    /l.sc = {/L.sc},
    /o.sc = {/O.sc},
    /oe.sc = {/OE.sc},
    /q.sc = {/Q.sc},
    /r.sc = {/R.sc},
    /t.sc = {/T.sc},
    /y.sc = {/Y.sc},
  % Cyrillic
    Г = {Ґ,Ғ,Ѓ,Ӷ},
    Ж = {Җ,Ӝ,Ӂ,Ж},
    З = {Ҙ,Ӟ},
    И = {Й,Ҋ,Ӣ,Ӥ,Ѝ},
    К = {Қ,Ҝ,Ҟ,Ҡ,Ќ,Ӄ},
    Л = {Ӆ,Ԓ,Ԯ},
    П = {Ԥ},
    У = {Ӯ,Ӱ,Ӳ,Ў},
    Ц = {Ꚑ,Ц},
    Ч = {Ҷ,Ҹ,Ӌ,Ӵ},
    Ш = {Щ},
    Ы = {Ӹ},
    Ь = {Ҍ},
    Э = {Ӭ},
    Ѵ = {Ѷ},
    Ҽ = {Ҿ̈},
    Ә = {Ӛ},
    г = {ѓ,ґ,ғ,ӷ},
    ж = {җ,ӂ,ӝ,ж},
    з = {ҙ,ӟ},
    и = {й,ѝ,ҋ,ӣ,ӥ},
    к = {ќ,қ,ҝ,ӄ}, % ҟ,ҡ
    л = {ӆ,ԓ,ԯ},
    м = {ӎ},
    н = {ң,ӈ,ӊ,ԩ}, %   ҥ
    п = {ԥ},
    т = {ҭ},
    ц = {ꚑ},
    ч = {ҷ,ҹ,ӌ,ӵ},
    ш = {щ},
    ы = {ӹ},
    э = {ӭ},
    ө = {ө,ӫ},
    ѵ = {ѷ},
    ү = {ұ},
    ҽ = {ҿ̈},
    ә = {ӛ},
  % Greek
    ϒ = {Ϋ,ϒ,ϔ,Ῠ,Ῡ},
 (l)ϒ = {Ύ,Ύ,Ὑ,Ὓ,Ὕ,Ὗ,Ὺ,Ύ},
 (l)Ω = {Ώ,Ὠ,Ὡ,Ὢ,Ὣ,Ὤ,Ὥ,Ὦ,Ὧ,ᾨ,ᾩ,ᾪ,ᾫ,ᾬ,ᾭ,ᾮ,ᾯ,Ὼ,Ώ},
    Ω = {ῼ,Ω}, % math
    Δ = {Δ}, % math
    Π = {Π}, % math
    α = {ά,ἀ,ἁ,ἂ,ἃ,ἄ,ἅ,ἆ,ἇ,ὰ,ά,ᾀ,ᾁ,ᾂ,ᾃ,ᾄ,ᾅ,ᾆ,ᾇ,ᾰ,ᾱ,ᾲ,ᾳ,ᾴ,ᾶ,ᾷ},
    ε = {έ,ἐ,ἑ,ἒ,ἓ,ἔ,ἕ,ὲ,έ},
    η = {ή,ἠ,ἡ,ἢ,ἣ,ἤ,ἥ,ἦ,ἧ,ὴ,ή,ῂ,ῃ,ῄ,ῆ,ῇ,ᾐ,ᾑ,ᾒ,ᾓ,ᾔ,ᾕ,ᾖ,ᾗ},
    ι = {ί,ϊ,ἰ,ἱ,ἲ,ἳ,ἴ,ἵ,ἶ,ἷ,ὶ,ί,ῐ,ῑ,ῒ,ΐ,ῖ,ῗ,ΐ},
    ο = {ό,ϙ,ὀ,ὁ,ὂ,ὃ,ὄ,ὅ,ὸ,ό},
      ρ = {ῤ,ῥ},
      υ = {ΰ,ϋ,ύ,ὐ,ὑ,ὒ,ὓ,ὔ,ὕ,ὖ,ὗ,ὺ,ύ,ῠ,ῡ,ῢ,ΰ,ῦ,ῧ},
      ω = {ώ,ὠ,ὡ,ὢ,ὣ,ὤ,ὥ,ὦ,ὧ,ὼ,ώ,ᾠ,ᾡ,ᾢ,ᾣ,ᾤ,ᾥ,ᾦ,ᾧ,ῲ,ῳ,ῴ,ῶ,ῷ},
    % other
      (1)  = {(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),(13),(14),(15),(16),(17),(18),(19),(20)},
      (a)  = {(b),(c),(d),(e),(f),(g),(h),(i),(j),(k),(l),(m),(n),(o),(p),(q),(r),(s),(t),(u),(v),(w),(x),(y),(z)},
      A  = {B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z},
      ! = {‼},
      ? = {⁇},
      . = {/onedotenleader},
      /endash = {/figuredash},
    }
⟨/EBGaramond⟩
```

3.1.4 Palatino

```
⟨*Palatino⟩
\DeclareCharacterInheritance
   { encoding = {TU,EU1,EU2},
     family = {Palatino} }
```

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-Palatino.cfg`.

```
   { A = {À,Á,Â,Ã,Ä,Ā,Ă,Ą,Ǎ,Ǟ,Ǡ,Ȧ,Ǻ,Ȁ,Ȃ,Ḁ,Ạ,Ả,Ấ,Ầ,Ẩ,Ẫ,Ậ,Ắ,Ằ,Ẳ,Ẵ,Ặ,Ꜳ},
     B = {Ḃ,Ḅ,Ḇ},
     C = {Ç,Ć,Ĉ,Ḉ,Ċ,Č},
     D = {Ď,Ḋ,Ḍ,Ḏ,Ḑ,Ḓ},
     E = {È,É,Ê,Ë,Ē,Ĕ,Ė,Ę,Ě,Ȅ,Ȩ,Ȇ,Ḕ,Ḗ,Ḙ,Ḛ,Ḝ,Ẹ,Ẻ,Ẽ,Ế,Ề,Ể,Ễ,Ệ},
     F = {Ḟ},
     G = {Ĝ,Ğ,Ġ,Ģ,Ǧ,Ǵ,Ḡ},
     H = {Ĥ,Ȟ,Ḣ,Ḥ,Ḧ,Ḩ,Ḫ},
     I = {Ì,Í,Î,Ȉ,Ï,Į,Ĩ,Ị,Ī,İ,Ǐ,Ỉ,Ȋ,Ĭ,Ḭ,Ḯ},
     J = {Ĵ},
     K = {Ķ,Ǩ,Ḱ,Ḳ,Ḵ},
     L = {Ĺ,Ļ,Ľ,Ḷ,Ḹ,Ḻ,Ḽ,Ỻ,Ŀ,Ł}, % L·
     M = {Ḿ,Ṁ,Ṃ},
     N = {Ñ,Ń,Ņ,Ň,Ǹ,Ṅ,Ṇ,Ṉ,Ṋ},
     O = {Ò,Ó,Ô,Õ,Ö,Ō,Ŏ,Ő,Ǒ,Ǫ,Ǭ,Ȍ,Ȏ,Ȫ,Ȭ,Ȯ,Ȱ,Ṍ,Ṏ,Ṑ,Ṓ,Ọ,Ỏ,Ố,Ồ,Ổ,Ỗ,Ộ,Ớ,Ờ,Ở,Ỡ,Ợ},
     P = {Ṕ,Ṗ},
     R = {Ŕ,Ŗ,Ř,Ȑ,Ȓ,Ṙ,Ṛ,Ṝ,Ṟ},
     S = {Ś,Ŝ,Ş,Š,Ș,Ṡ,Ṣ,Ṥ,Ṧ,Ṩ},
     T = {Ţ,Ť,Ț,Ṫ,Ṭ,Ṯ,Ṱ},
     U = {Ù,Ú,Û,Ü,Ũ,Ū,Ŭ,Ů,Ű,Ų,Ǔ,Ǖ,Ǘ,Ǚ,Ǜ,Ȕ,Ȗ,Ṳ,Ṵ,Ṷ,Ṹ,Ṻ,Ụ,Ủ,Ứ,Ừ,Ử,Ữ,Ự},
     V = {Ṽ,Ṿ},
     W = {Ŵ,Ẁ,Ẃ,Ẅ,Ẇ,Ẉ},
     X = {Ẋ,Ẍ},
     Y = {Ý,Ŷ,Ÿ,Ȳ,Ẏ,Ỳ,Ỵ,Ỷ,Ỹ},
     Z = {Ź,Ż,Ž,Ẑ,Ẓ,Ẕ},
     a = {à,á,â,ã,ä,å,ā,ă,ą,ǎ,ǟ,ǡ,ȧ,ǻ,ȁ,ȃ,ḁ,ạ,ả,ầ,ấ,ẩ,ẫ,ậ,ắ,ằ,ẳ,ẵ,ặ}, % aᵓ
     b = {ḃ,ḅ,ḇ},
     c = {ç,ć,ĉ,ċ,č,ḉ},
     d = {ď,ḋ,ḍ,ḏ,ḑ,ḓ},
     e = {è,é,ê,ë,ē,ĕ,ė,ę,ě,ȅ,ȩ,ȇ,ḕ,ḗ,ḙ,ḛ,ḝ,ẹ,ẻ,ẽ,ề,ế,ể,ễ,ệ},
     f = {ḟ,ﬀ},
     g = {ĝ,ğ,ġ,ģ,ǧ,ǵ,ḡ},
     h = {ĥ,ȟ,ḣ,ḥ,ḧ,ḩ,ḫ,ẖ},
     i = {ı,ì,í,î,ȉ,ǐ,ï,į,ī,ị,ĩ,ḭ,ḯ,ỉ,ȋ,ĭ},
     j = {ĵ,ǰ},
     k = {ķ,ǩ,ḱ,ḳ,ḵ},
     l = {ĺ,ļ,ḻ,ḷ,ḹ,ḽ}, % ľ,l·
```

```
    m = {ḿ,ṁ,ṃ},
    n = {ñ,ń,ņ,ň,ǹ,ṅ,ṇ,ṉ,ṋ}, % ŉ
    o = {ò,ó,ô,õ,ö,ō,ŏ,ő,ǒ,ǫ,ǭ,ȍ,ȏ,ȫ,ȭ,ȯ,ȱ,ṍ,ṏ,ṑ,ṓ,ọ,ỏ,ồ,ố,ổ,ỗ,ộ,ớ,ờ,ở,ỡ,ợ},
    p = {ṕ,ṗ},
    r = {ŕ,ŗ,ř,ȑ,ȓ,ṙ,ṛ,ṝ,ṟ},
    s = {ś,ŝ,ş,š,ș,ṡ,ṣ,ṥ,ṧ,ṩ},
    t = {ţ,ț,ṫ,ṭ,ṯ,ṱ,ẗ}, % ť
    u = {ù,ú,û,ü,ũ,ū,ŭ,ů,ű,ų,ǔ,ǖ,ǘ,ǚ,ǜ,ȕ,ȗ,ṳ,ṵ,ṷ,ṹ,ṻ,ụ,ủ,ứ,ừ,ử,ữ,ự},
    v = {ṽ,ṿ},
    w = {ŵ,ẁ,ẃ,ẅ,ẇ,ẉ,ẘ},
    x = {ẋ,ẍ},
    y = {ý,ÿ,ŷ,ȳ,ẏ,ẙ,ỳ,ỵ,ỷ,ỹ},
    z = {ź,ż,ž,ẑ,ẓ,ẕ},
  }
⟨/Palatino⟩
```

3.1.5 Basic glyph set

There are quite a few fonts out there that don't even fill the T1 glyph set. To prevent a plethora of warnings, they may be aliased to the surrogate font TU-basic. Examples of such fonts are: Lato, Fontin and Bergamo.

```
⟨*TU-basic⟩
\DeclareCharacterInheritance
  { encoding = {TU,EU1,EU2},
    family   = {TU-basic} }
  { A = {À,Á,Â,Ã,Ä,Å},
    a = {à,á,â,ã,ä,å},
    C = {Ç},
    c = {ç},
    D = {Đ},
    E = {È,É,Ê,Ë},
    e = {è,é,ê,ë},
    I = {Ì,Í,Î,Ï},
    i = {ì,í,î,ï,ı},
    L = {Ł},
    l = {ł},
    N = {Ñ},
    n = {ñ},
    O = {Ø,Ò,Ó,Ô,Õ,Ö},
    o = {ø,ò,ó,ô,õ,ö},
    S = {Š},
    s = {š},
    U = {Ù,Ú,Û,Ü},
    u = {ù,ú,û,ü},
    Y = {Ý,Ÿ},
```

For some reason, the ÿ in the next line comes out as ß. Don't worry, there's really a `y diaeresis` in the source.

```
    y = {ý,ß},
    Z = {Ž},
    z = {ž}
  }
⟨/TU-basic⟩
```

3.1.6 Empty glyph set

Other fonts, e.g., the self-professedly awesone Font Awesome font, have no meaningful glyph arsenal at all, and should therefore be aliased so that empty settings are applied.

```
⟨*TU-empty⟩
\DeclareCharacterInheritance
```

```
  { encoding = {TU,EU1,EU2},
    family   = {TU-empty} }
  { }
⟨/TU-empty⟩
```

3.2 Character protrusion

```

%%% -----------------------------------------------------------------------
%%% PROTRUSION

```

3.2.1 Latin Modern Roman/New Computer Modern

```
⟨*LatinModernRoman|NewComputerModern⟩
\SetProtrusion
⟨LatinModernRoman⟩   [ name     = LMR-default ]
⟨NewComputerModern⟩   [ name     = NCM-default ]
⟨LatinModernRoman⟩   { encoding = {TU,EU1,EU2},
⟨LatinModernRoman⟩     family   = Latin Modern Roman }
⟨NewComputerModern⟩   { }
   {
    A = {50,50},
    Æ = {50, },
    F = { ,50},
    J = {50, },
    K = { ,50},
    L = { ,50},
    T = {50,50},
    V = {50,50},
    W = {50,50},
    X = {50,50},
    Y = {50,50},
    k = { ,50},
    r = { ,50},
    t = { ,70},
    v = {50,50},
    w = {50,50},
    x = {50,50},
    y = {50,70},
    0 = { ,50},
    1 = {100,200},
    2 = {50,50},
    3 = {50,50},
    4 = {70,70},
    5 = { ,50},
    6 = { ,50},
    7 = {50,100},
    8 = { ,50},
    9 = { ,50},
    . = { ,700},
   {,}= { ,500},
    : = { ,500},
    ; = { ,500},
    ! = { ,100},
    ? = { ,200},
    @ = {50,50},
    ~ = {200,250},
   \% = {50,50},
    * = {300,300},
    + = {250,250},
    - = {400,500},  % /hyphen
    – = {400,300},  % /endash
    — = {300,200},  % /emdash
    _ = {200,200},  % /underscore
    / = {200,300},
   /backslash    = {200,300},
    ' = {300,400}, % /quotesingle
    ‘ = {300,400},  ’ = {300,400},
    “ = {300,300},  ” = {300,300},
    , = {400,400},  „ = {400,400},
    ‹ = {400,400},  › = {300,500},
    « = {300,200},  » = {100,400},
    ¡ = {100,   },  ¿ = {100,   },
    ( = {300,   },  ) = {   ,300},
    <  = {200,100}, > = {100,200},
   /braceleft = {400,200}, /braceright = {200,400},
   /angleleft = {400,   }, /angleright = {   ,400},
    † = {100,100},
    ‡ = { 80, 80},
    • = {200,200},
    · = {400,450}, % / periodcentered
    ℃ = { 80, 50},
    ₡ = {   , 50},
    ° = {400,400},
    ™ = {100,200},
    © = {100,100},
    ® = {100,100},
    ª = {100,200},
    º = {100,200},
    ¹ = {200,250},
    ² = { 50,100},
    ³ = { 50,100},
    ¬ = {200,   },
    − = {300,300},
    ± = {150,200},
    × = {150,250},
    ÷ = {150,250},
    € = {100,   },
⟨*LatinModernRoman⟩
   /one.oldstyle   = {100,100},
   /two.oldstyle   = { 50, 50},
   /three.oldstyle = { 30, 80},
   /four.oldstyle  = { 50, 50},
   /seven.oldstyle = { 50, 80},
⟨/LatinModernRoman⟩
⟨*NewComputerModern⟩
    A = {50,50},   % /Alphatonos
    Ὰ = {120,50},  %
    Ἀ = {120,50},  %
    Ἁ = {80,50},  %
    Ἂ = {220,50}, %
    Ἃ = {220,50}, %
    Ἄ = {170,50}, %
    Ἅ = {170,50}, %
    Ἆ = {190,50}, %
    Ἇ = {190,50}, %
    ᾈ = {150,50}, %
    ᾉ = {80,50},  %
    ᾊ = {220,50}, %
    ᾋ = {220,50}, %
    ᾌ = {170,50}, %
    ᾍ = {170,50}, %
    ᾎ = {210,50}, %
    ᾏ = {210,50}, %
    /uni1FBC.alt = {,205}, % Alpha prosgegrammeni
    /uni1F88.alt = {50,190}, %Alpha psili prosgegrammeni
    /uni1F89.alt = {,200}, %Alpha dasia prosgegrammeni
    /uni1F8A.alt = {130,180}, %Alpha psili baria prosgegrammeni
    /uni1F8B.alt = {130,190}, %Alpha dasia baria prosgegrammeni
```

```
    /uni1F8C.alt = {100,190}, %Alpha psili oxia prosgegrammeni
    /uni1F8D.alt = {70,190}, %Alpha dasia oxia prosgegrammeni
    /uni1F8E.alt = {120,190}, %Alpha psili perispomeni prosgegrammeni
    /uni1F8F.alt = {120,190}, %Alpha dasia perispomeni prosgegrammeni
  %
    /uni1FCC.alt = {,205}, % Eta prosgegrammeni
    /uni1F98.alt = {185,170}, %Eta psili prosgegrammeni
    /uni1F99.alt = {185,170}, %Eta dasia prosgegrammeni
    /uni1F9A.alt = {220,170}, %Eta psili baria prosgegrammeni
    /uni1F9B.alt = {220,170}, %Eta dasia baria prosgegrammeni
    /uni1F9C.alt = {220,170}, %Eta psili oxia prosgegrammeni
    /uni1F9D.alt = {220,170}, %Eta dasia oxia prosgegrammeni
    /uni1F9E.alt = {255,170}, %Eta psili perispomeni prosgegrammeni
    /uni1F9F.alt = {255,170}, %Eta dasia perispomeni prosgegrammeni
  %
    O = {95,50}, %
⟨/NewComputerModern⟩
    Γ = {    ,180}, % /Gamma
⟨LatinModernRoman⟩     Δ = {100,100}, % /Delta
⟨NewComputerModern⟩     Δ = {50,50}, % /Delta
    Θ = { 50, 50}, % /Theta
⟨LatinModernRoman⟩     Λ = {100,100}, % /Lambda
⟨NewComputerModern⟩     Λ = {50,50}, % /Lambda
%    Ξ = {,},        % /Xi
%    Π = {,},        % /Pi
    Σ = { 50, 50}, % /Sigma
⟨LatinModernRoman⟩     ϒ = {100,100}, % /Upsilon
⟨NewComputerModern⟩     ϒ = {80,80}, % /Upsilon
    Φ = { 50, 50}, % /Phi
    Ψ = { 50, 50}, % /Psi
⟨*NewComputerModern⟩
    Ω = { 20, 30}, % /Omega
    Ω = {150,30},
    Ὠ = {220,30},
    Ὡ = {205,30},
    Ὢ = {285,30},
    Ὣ = {285,30},
    Ὤ = {270,30},
    Ὥ = {270,30},
    Ὦ = {310,30},
    Ὧ = {310,30},
    ᾨ = {205,30},
    ᾩ = {205,30},
    ᾪ = {285,30},
    ᾫ = {285,30},
    ᾬ = {270,30},
    ᾭ = {270,30},
    ᾮ = {310,30},
    ᾯ = {310,30},
    /uni1FFC.alt = {,230}, % Omega prosgegrammeni
    /uni1FA8.alt = {185,190}, %Omega psili prosgegrammeni
    /uni1FA9.alt = {185,190}, %Omega dasia prosgegrammeni
    /uni1FAA.alt = {220,190}, %Omega psili baria prosgegrammeni
    /uni1FAB.alt = {220,190}, %Omega dasia baria prosgegrammeni
    /uni1FAC.alt = {220,190}, %Omega psili oxia prosgegrammeni
    /uni1FAD.alt = {220,190}, %Omega dasia oxia prosgegrammeni
    /uni1FAE.alt = {255,190}, %Omega psili perispomeni prosgegrammeni
    /uni1FAF.alt = {255,190}, %Omega dasia perispomeni prosgegrammeni
  %
    α = {,50},
    γ = {50,50},
    ζ = {,50},
    ϑ = {30,40},
    ι = {,50},
    ϊ = {-20,-30},
    ϰ = {50,50},
    λ = {50,50},
    ν = {50,25},
    π = {50,50},
    σ = {,50},
    ς = {,50},
    τ = {50,50},
    χ = {50,50},
    ψ = {50,50},
%     /uni1F98.alt = {,},
```

CMU Serif doesn't include *.end glyphs, and the OldStyle numbers' names differ.

```
    }

\SetProtrusion
    [ name     = NCM-TU,
      load     = NCM-default ]
    { encoding = {TU,EU1,EU2},
      family   = {New Computer Modern} }
    {
      /a.end = {,330},
      /e.end = {,350},
      /k.alt = { ,50},
      /r.end = {,300},
      /m.end = {,200},
      /n.end = {,300},
      /one.oldstyle   = {100,100},
      /two.oldstyle   = { 50, 50},
      /three.oldstyle = { 30, 80},
      /four.oldstyle  = { 50, 50},
      /seven.oldstyle = { 50, 80},
    }

\SetProtrusion
    [ name     = CMU-TU,
      load     = NCM-default ]
    { encoding = {TU,EU1,EU2},
      family   = {CMU Serif} }
    {
      /oneoldstyle   = {100,100},
      /twooldstyle   = { 50, 50},
      /threeoldstyle = { 30, 80},
      /fouroldstyle  = { 50, 50},
      /sevenoldstyle = { 50, 80},
⟨/NewComputerModern⟩
    }

\SetProtrusion
⟨LatinModernRoman⟩   [ name     = LMR-it ]
⟨NewComputerModern⟩   [ name     = NCM-it ]
⟨LatinModernRoman⟩   { encoding = {TU,EU1,EU2},
⟨LatinModernRoman⟩     family   = Latin Modern Roman,
⟨LatinModernRoman⟩     shape    = {it,sl}        }
⟨NewComputerModern⟩   { }
    {
    A = {125,100},
    Æ = {125,-55},
    B = {90,-40},
    C = {145,-75},
    D = {75, -28},
    E = {80,-55},
    F = {85,-80},
    G = {153,-15},
    H = {73,-60},
    I = {140,-120},
    IJ = {140,-80},
    J = {135,-80},
    K = {70,-30},
    L = {87, 40},
    M = {67,-45},
    N = {75,-55},
    O = {150,-30},
    Œ = {150,-55},
    P = {82,-50},
    Q = {150,-30},
    R = {75, 15},
    S = {90,-65},
    $ = {100,-20},
    T = {220,-85},
    U = {230,-55},
    V = {260,-60},
    W = {185,-55},
    X = {70,-30},
    Y = {250,-60},
    Z = {90,-60},
    a = {150,-10},
    b = {170,    },
    c = {173,-10},
    d = {150,-55},
    e = {180, },
    f = {   ,-250},
    g = {150,-10},
    h = {100, },
    i = {210, },
    ij = {210,-40},
    j = {   ,-40},
    k = {110,-50},
    l = {240,-110},
    m = {80, },
    n = {115, },
    o = {155, },
    q = {170,-40},
    r = {155,-40},
    s = {130, },
    t = {230,-10},
    u = {120, },
    v = {140,-25},
    w = {98,-20},
    x = {65,-40},
    y = {130,-20},
    z = {110,-80},
    0 = {170,-85},
    1 = {230,110},
    2 = {130,-70},
    3 = {140,-70},
    4 = {130,80},
    5 = {160, },
    6 = {175,-30},
    7 = {250,-150},
    8 = {130,-40},
    9 = {155,-80},
    . = { ,500},
   {,}= { ,450},
    : = { ,300},
    ; = { ,300},
    & = {130,30},
   \% = {180,50},
    * = {380,20},
    + = {180,200},
    @ = {180,10},
    ~ = {200,150},
    ( = {300, },    ) = {  ,70},
    / = {100,100},
    - = {500,300}, % /hyphen
    – = {500,300}, % /endash
    — = {400,170}, % /emdash
    _ = {100,200}, % /underscore
    ' = {300,400}, % /quotesingle
    " = {500,300},
    ‘ = {800,200},  ’ = {800,-20},
    “ = {540,100},  ” = {500,100},
    , = {300,700},  „ = {200,600},
    ‹ = {500,300},  › = {400,400},
    « = {400,100},  » = {200,300},
    ¡ = {200,   },  ¿ = {200,   },
    < = {300,100},  > = {200,100},
   /backslash   = {300,300},
   /braceleft   = {400,100},   /braceright    = {200,200},
    † = {200, 80},
    ‡ = {120, 80},
    • = {220,100},
    · = {550,300}, % / periodcentered
    ℃ = {170,   },
    ₡ = {100, 50},
    ¶ = {200,   },
    ° = {500,300},
    ™ = {200, 70},
    © = { 50, 70},
    ® = { 50, 70},
    ª = {140,100},
    º = {140,100},
    ¹ = {400,150},
    ² = {250, 80},
    ³ = {250, 80},
    ¬ = {250, 80},
    − = {300,200},
    ± = {150,170},
    × = {200,200},
    ÷ = {200,200},
    € = {150,   },
⟨*LatinModernRoman⟩
   /one.oldstyle   = {100,100},
   /two.oldstyle   = {100, 80},
   /three.oldstyle = { 80, 50},
   /four.oldstyle  = { 80, 80},
   /five.oldstyle  = { 50,   },
   /six.oldstyle   = { 50,   },
   /seven.oldstyle = { 80, 80},
   /eight.oldstyle = { 50,   },
⟨/LatinModernRoman⟩
    Γ = {100,120}, % /Gamma
    Δ = {120,100}, % /Delta
    Θ = {120, 50}, % /Theta
⟨LatinModernRoman⟩     Λ = {130,100}, % /Lambda
⟨NewComputerModern⟩     Λ = {160,100}, % /Lambda
    Ξ = {100,},    % /Xi
    Π = {100,},    % /Pi
    Σ = {100, 50}, % /Sigma
⟨LatinModernRoman⟩     ϒ = {180,100}, % /Upsilon
⟨NewComputerModern⟩     ϒ = {260,100}, % /Upsilon
    Φ = {130, 70}, % /Phi
    Ψ = {130, 50}, % /Psi
    Ω = { 50,},    % /Omega
⟨*NewComputerModern⟩
    Ά = {190,50},  %
```

```
  Ἀ = {220,50},  %
  Ἁ = {200,50},  %
  Ἂ = {300,50}, %
  Ἃ = {300,50}, %
  Ἄ = {300,50}, %
  Ἅ = {300,50}, %
  Ἆ = {320,50}, %
  Ἇ = {320,50}, %
  ᾈ = {200,50}, %
  ᾉ = {200,50},  %
  ᾊ = {300,50}, %
  ᾋ = {300,50}, %
  ᾌ = {300,50}, %
  ᾍ = {300,50}, %
  ᾎ = {320,50}, %
  ᾏ = {320,50}, %
  /uni1FBC.alt = {,205}, % Alpha prosgegrammeni
  /uni1F88.alt = {50,190}, %Alpha psili prosgegrammeni
  /uni1F89.alt = {,200}, %Alpha dasia prosgegrammeni
  /uni1F8A.alt = {130,180}, %Alpha psili baria prosgegrammeni
  /uni1F8B.alt = {130,190}, %Alpha dasia baria prosgegrammeni
  /uni1F8C.alt = {100,190}, %Alpha psili oxia prosgegrammeni
  /uni1F8D.alt = {70,190}, %Alpha dasia oxia prosgegrammeni
  /uni1F8E.alt = {120,190}, %Alpha psili perispomeni prosgegrammeni
  /uni1F8F.alt = {120,190}, %Alpha dasia perispomeni prosgegrammeni
%
  /uni1FCC.alt = {,205}, % Eta prosgegrammeni
  /uni1F98.alt = {185,170}, %Eta psili prosgegrammeni
  /uni1F99.alt = {185,170}, %Eta dasia prosgegrammeni
  /uni1F9A.alt = {220,170}, %Eta psili baria prosgegrammeni
  /uni1F9B.alt = {220,170}, %Eta dasia baria prosgegrammeni
  /uni1F9C.alt = {220,170}, %Eta psili oxia prosgegrammeni
  /uni1F9D.alt = {220,170}, %Eta dasia oxia prosgegrammeni
  /uni1F9E.alt = {255,170}, %Eta psili perispomeni prosgegrammeni
  /uni1F9F.alt = {255,170}, %Eta dasia perispomeni prosgegrammeni
%
  O = {95,50}, %
  Ω = {120, 30}, % /Omega
  Ω = {160,30},
  Ὠ = {250,30},
  Ὡ = {250,30},
  Ὢ = {300,30},
  Ὣ = {300,30},
  Ὤ = {300,30},
  Ὥ = {300,30},
  Ὦ = {330,30},
  Ὧ = {330,30},
  ῼ = {30,30},
  ᾨ = {230,30},
  ᾩ = {230,30},
  ᾪ = {300,30},
  ᾫ = {300,30},
  ᾬ = {300,30},
  ᾭ = {300,30},
  ᾮ = {330,30},
  ᾯ = {330,30},
  /uni1FFC.alt = {,230}, % Omega prosgegrammeni
  /uni1FA8.alt = {185,190}, %Omega psili prosgegrammeni
  /uni1FA9.alt = {185,190}, %Omega dasia prosgegrammeni
  /uni1FAA.alt = {220,190}, %Omega psili baria prosgegrammeni
  /uni1FAB.alt = {220,190}, %Omega dasia baria prosgegrammeni
  /uni1FAC.alt = {220,190}, %Omega psili oxia prosgegrammeni
  /uni1FAD.alt = {220,190}, %Omega dasia oxia prosgegrammeni
  /uni1FAE.alt = {255,190}, %Omega psili perispomeni prosgegrammeni
  /uni1FAF.alt = {255,190}, %Omega dasia perispomeni prosgegrammeni
  %
    α = {50,50},
    γ = {100,50},
    δ = {30,50},
    ε = {30,},
    ζ = {20,50},
    ϑ = {30,40},
    ι = {,50},
    ϊ = {-20,-30},
    ϰ = {50,50},
    λ = {-20,50},
    ν = {50,25},
    ο ={40,},
    π = {50,50},
    σ = {40,50},
    ς = {20,50},
    τ = {50,50},
    υ = {80,},
    φ = {80,},
    χ = {20,},
    ψ = {80,},
%    /uni1F98.alt = {,},
    }

\SetProtrusion
    [ name     = NCM-it-TU,
      load     = NCM-it ]
    { encoding = {TU,EU1,EU2},
      family   = {New Computer Modern},
      shape    = {it,sl} }
    {
      /a.end = {,330}, %Fix
      /e.end = {,350}, %Fix
      /k.alt = {  ,50}, %Fix
      /r.end = {,300}, %Fix
      /m.end = {,200}, %Fix
      /n.end = {,300}, %Fix
      /one.oldstyle   = {100,100},
      /two.oldstyle   = {100, 80},
      /three.oldstyle = { 80, 50},
      /four.oldstyle  = { 80, 80},
      /five.oldstyle  = { 50,   },
      /six.oldstyle   = { 50,   },
      /seven.oldstyle = { 80, 80},
      /eight.oldstyle = { 50,   },
    }

\SetProtrusion
    [ name     = CMU-it-TU,
      load     = NCM-it ]
    { encoding = {TU,EU1,EU2},
      family   = {CMU Serif},
      shape    = {it,sl} }
    {
      /oneoldstyle   = {100,100},
      /twooldstyle   = {100, 80},
      /threeoldstyle = { 80, 50},
      /fouroldstyle  = { 80, 80},
      /fiveoldstyle  = { 50,   },
      /sixoldstyle   = { 50,   },
      /sevenoldstyle = { 80, 80},
      /eightoldstyle = { 50,   },
⟨/NewComputerModern⟩
    }
⟨/LatinModernRoman|NewComputerModern⟩
```

3.2.2 Charis SIL

```
⟨*CharisSIL⟩
\SetProtrusion
   [ name     = Charis-default ]
   { encoding = {TU,EU1,EU2},
     family   = Charis SIL }
   {
      A  =  {50,50},
      Æ  =  {50,50},
      C  =  {50,   },
      D  =  {   ,50},
      F  =  {   ,50},
      G  =  {50,   },
      J  =  {100,  },
      K  =  {   ,50},
      L  =  {   ,50},
      L·  =  {   ,100},
      O  =  {50,50},
      Œ  =  {50,   },
      P  =  {   ,50},
      Q  =  {50,70},
      R  =  {   ,50},
      ẞ  =  {   ,40},  % capital sharp s
      T  =  {50,50},
      V  =  {50,50},
      W  =  {50,50},
      X  =  {50,50},
      Y  =  {50,50},
      k  =  {   ,50},
      l·  =  {   ,150},
      r  =  {   ,50},
      t  =  {   ,50},
      v  =  {50,50},
      w  =  {50,50},
      x  =  {50,50},
      y  =  {   ,50},
      1  =  {150,150},
      2  =  {50,50},
      3  =  {50,   },
      4  =  {100,50},
      6  =  {50,   },
      7  =  {50,80},
      9  =  {50,50},
      .  =  {  ,600},
     {,}=  {  ,500},
      :  =  {  ,400},
      ;  =  {  ,300},
      !  =  {  ,100},
      ?  =  {  ,200},
      @  =  {50,50},
      ~  =  {200,250},
     \%  =  {   ,50},
      *  =  {300,300},
      +  =  {200,250},
      /  =  {  ,200},
     /backslash  =  {150,200},
      |  =  {200,200},
      -  =  {400,500},  % hyphen
      –  =  {200,300},  % endash
      —  =  {150,250},  % emdash
      ―  =  {200,200},  % Horizontal Bar  =  \texttwelveudash
      ‒  =  {150,150},  % Figure Dash  =  \textthreequartersemdash
      _  =  {100,100},
     {=}  =  {100,100},
      ‘ = {300,400},   ’ = {300,400},
      “ = {300,300},   ” = {300,300},
      , = {400,400},   „ = {300,300},
      ‹ = {400,300},   › = {300,400},
      « = {200,200},   » = {150,300},
      ¡ = {100,   },   ¿ = {100,   },
      ( = {200,   },   ) = {   ,200},
      < = {200,150},   > = {100,200},
      [ = {100,   },   ] = {   ,100},
     /braceleft = {200,   }, /braceright = {   ,300},
      † = { 80, 80},
      ‡ = {100,100},
      • = {200,200},
      ° = {150,200},
      ™ = {150,150},
      ¢ = { 50,   },
      £ = { 50,   },
      ¦ = {200,200},
      © = {100,100},
      ® = {100,100},
      ª = {100,200},
      º = {200,200},
      ¬ = {200, 50},
      µ = {   ,100},
      ¶ = {   ,100},
      · = {300,400},
      ¹ = {200,300},
      ² = {100,200},
      ³ = {100,200},
      € = {100,   },
      ± = {150,200},
      × = {200,200},
      ÷ = {250,250},
     /minus = {200,200},
      − = {200,200},
     % Cyrillic
      Б = {  ,50},
      Г = {  ,130},
      Ж = {50,50},
      З = {30,50},
      Л = {50,  },
      У = {50,50},
      Ф = {50,50},
      Ч = {100, },
      Ъ = {  ,50},
      Ь = {  ,50},
      Э = {50,50},
      Ю = {  ,40},
      Я = {50,  },
      Ѵ = {50,50},
      Ҽ = {50,  },
      Ђ = {50,100},
      Є = {50,  },
      Љ = {50,50},
      Њ = {  ,50},
      Ћ = {50,50},
      Ԅ = {100,100},
      Ԇ = {50,50},
      Ѣ = {  ,50},
      Ҍ = {  ,50},
      Ԉ = {50,80},
      Ԋ = {  ,80},
      Ԏ = {50,50},
      Ԓ = {50,  },
      Ԕ = {50,40},
      Ҡ = {   ,50},
      Ӕ = {50,   },
      Ԡ = {   ,50},
      Ԣ = {   ,50},
      Ԃ = {   ,100},
      б = {50,50},
      г = {   ,70},
      к = {   ,50},
      л = {50,   },
      т = {50,50},
      ф = {50,50},
      ч = {50,   },
      ъ = {   ,50},
      ь = {   ,50},
      э = {   ,50},
      я = {50,   },
      љ = {50,   },
      њ = {   ,50},
      ѣ = {   ,50},
      ѵ = {50,50},
      ҽ = {50,   },
      ҍ = {   ,50},
      ү = {50,50},
      ҕ = {   ,50},
      ҧ = {   ,50},
      ԃ = {   ,100},
      ԅ = {100,100},
      ԇ = {50,50},
      ԉ = {50,70},
      ԋ = {   ,70},
      ӕ = {50,30},
      ԡ = {   ,50},
      ԣ = {   ,50},
      %      Д П Ц Ш Щ Ы Ҕ Ҧ Ѡ Ә Ҵ Џ Ӭ Ӡ Ԑ Ԁ
      %      в д ж з и м н п ц ш ы ю ђ є ћ џ ә ԑ ѡ ҵ ӡ ԁ ԍ ԏ ԓ ԕ ԗ
    % Greek
      Δ = {50,50},
      Ψ = {50,50},
      γ = {70,70},
      λ = {40,70},
      π = {40,50},
      ρ = {   ,50},
      σ = {   ,50},
      χ = {50,50},
}

\SetProtrusion
   [ name     = Charis-it   ]
   { encoding = {TU,EU1,EU2},
     family   = Charis SIL,
     shape    = {it,sl}  }
   {
      C = {50,   },
      G = {50,   },
      J = {50,   },
      L = {50,50},
      O = {50,   },
      Œ = {50,   },
      Q = {50,   },
      S = {50,   },
      $ = {50,   },
      T = {70,   },
      o = {50,50},
      p = {   ,50},
      q = {50,   },
   t  =  {    ,50},
   w  =  {    ,50},
   y  =  {    ,50},
   1  =  {150,100},
   3  =  {50,    },
   4  =  {100,  },
   6  =  {50,    },
   7  =  {100,  },
   .  =  {  ,700},
  {,}=  {  ,600},
   :  =  {  ,400},
   ;  =  {  ,400},
   ?  =  {  ,150},
   &  =  {    ,80},
  \%  =  {50,50},
   *  =  {300,200},
   +  =  {250,250},
   @  =  {80,50},
   ~  =  {150,150},
   /  =  {    ,150},
  /backslash  =  {150,150},
   -  =  {300,400},  %  hyphen
   –  =  {200,300},  %  endash
   —  =  {150,200},  %  emdash
   _  =  {    ,100},
  {=}  =  {200,200},
   ±  =  {150,200},
   ×  =  {250,250},
   ÷  =  {250,250},
   °  =  {150,200},
   ·  =  {300,400},
   ‘  =  {400,200},    ’  =  {400,200},
   “  =  {300,200},    ”  =  {400,200},
   ‚  =  {200,500},    „  =  {150,500},
   ‹  =  {300,400},    ›  =  {200,500},
   «  =  {200,300},    »  =  {150,400},
   (  =  {200,      },    )  =  {      ,200},
   <  =  {200,200},    >  =  {200,200},
  /braceleft  =  {300,      },  /braceright  =  {      ,200},
% Cyrillic
   Ж  =  {50,30},
   Л  =  {50,    },
   У  =  {50,30},
   Ф  =  {50,    },
   Ч  =  {100,  },
   Ъ  =  {    ,50},
   Ь  =  {    ,50},
   Э  =  {50,50},
   Я  =  {50,    },
   Ѵ  =  {50,50},
   Љ  =  {50,50},
   Ꙉ  =  {140,100},
   Ꙃ  =  {70,50},
   Љ  =  {50,80},
   Њ  =  {    ,80},
   Ћ  =  {50,50},
   г  =  {50,50},
   д  =  {50,30},
   м  =  {50,    },
   ф  =  {50,    },
   ч  =  {50,    },
   ъ  =  {    ,50},
   ь  =  {    ,50},
   э  =  {    ,50},
   я  =  {50,    },
      љ  =  {50,50},
      њ  =  {   ,50},
      ѵ  =  {50,50},
      ѣ  =  {   ,50},
      ԅ  =  {140,100},
      ԇ  =  {70,50},
      ԉ  =  {50,70},
      ԋ  =  {   ,70},
    %  Greek
      Γ  =  {   ,130},
      Δ  =  {50,50},
      Ψ  =  {50,50},
      γ  =  {70,70},
      λ  =  {40,70},
      π  =  {40,50},
      ρ  =  {   ,50},
      σ  =  {   ,50},
      χ  =  {50,50},
  }
```

The small caps glyph names in Charis SIL have changed with version 5.0 of the font. We try to get the names right both with LuaTEX (where we can simply query the font version) and with XƎTEX (where we check for glyph name).

```

 % quick and dirty -- maybe we'll promote this to a
 % regular key some time
\define@key{MT@pr@c}{command}{\csname #1\endcsname}

 % glyph names have changed with version 5.0 of Charis SIL:
 % before: /a.SC, /b.SC, ...
 % after:  /a.sc, /b.sc, ...
\ifx\MT@lua\@undefined
  \gdef\MT@get@CHARIS@SC{
    % test whether glyph "a.sc" exists
    \ifnum\numexpr\XeTeXglyphindex "a.sc"\relax > 0
      \gdef\MT@CHARIS@SC{sc}%
    \else
      \gdef\MT@CHARIS@SC{SC}%
    \fi
  }
\else
  \gdef\MT@get@CHARIS@SC{
    \gdef\MT@CHARIS@SC{\MT@lua{
      % check font version
% --  why doesn't this work?:
%     f = font.getfont(font.current());
%     i = fontloader.info(f.filename);
%     if (tonumber(i.version) < 5) then;
      if (tonumber(fontloader.info(font.getfont(font.current()).filename).version) < 5) then;
        tex.print("SC");
      else;
        tex.print("sc");
      end
    }}
  }
\fi

\SetProtrusion
   [ name     = Charis-sc,
     load     = Charis-default,
     command  = {MT@get@CHARIS@SC} ]
   { encoding = {TU,EU1,EU2},
     family   = Charis SIL,
     shape    = {sc}  }
    {
%      A = {100,100}, % etc., doesn't work with \textsc
      /a.\MT@CHARIS@SC = {100,100},
      /c.\MT@CHARIS@SC = {50,  },
      /d.\MT@CHARIS@SC = {  ,50},
      /f.\MT@CHARIS@SC = {  ,50},
      /g.\MT@CHARIS@SC = {50,  },
      /j.\MT@CHARIS@SC = {100, },
      /k.\MT@CHARIS@SC = {  ,50},
      /l.\MT@CHARIS@SC = {  ,50},
    /f_l.\MT@CHARIS@SC = {  ,50},
      /o.\MT@CHARIS@SC = {50,50},
     /oe.\MT@CHARIS@SC = {50,  },
      /q.\MT@CHARIS@SC = {50,70},
      /r.\MT@CHARIS@SC = {  ,50},
      /t.\MT@CHARIS@SC = {50,100},
      /v.\MT@CHARIS@SC = {50,50},
      /w.\MT@CHARIS@SC = {50,50},
      /x.\MT@CHARIS@SC = {50,50},
      /y.\MT@CHARIS@SC = {50,50}
    }
⟨/CharisSIL⟩
```

3.2.3 EB Garamond

```
⟨*EBGaramond⟩
\SetProtrusion
   [ name     = EBGaramond-TU,
     load     = EBGaramond-T1-LF ]
   { encoding = {TU,EU1,EU2},
     family   = EBGaramond }
   {
    /one.tosf   = {150,150},
    /two.tosf   = {50,50},
    /three.tosf = {50,50},
    /four.tosf  = {50,50},
    /five.tosf  = {50,50},
    /six.tosf   = {50,50},
    /seven.tosf = {50,80},
    /eight.tosf = {50,50},
    /nine.tosf  = {50,50},
    /one.lf     = {50,50},
    /two.lf     = {50,50},
    /four.lf    = {50,50},
    /seven.lf   = {50,50},
    /one.osf    = {50,50},
    /two.osf    = {50,50},
    /four.osf   = {50,50},
    /seven.osf  = {50,50},

    IV =  {    , 35},
    VI =  { 35,    },
    VII   = { 30,    },
    VIII   = { 25,    },
    IX =  {    , 35},
    XI =  { 35,    },
    XII =  { 30,    },
    iv =  {    , 25},
    vi =  { 25,    },
    vii   = { 20,    },
    viii   = { 20,    },
    ix =  {    , 25},
    xi =  { 25,    },
    xii   = { 20,    },
  % textcomp
   \textquotesingle = {400,500},
    ˷ = {200,250},
    ƒ = {    ,100},
    ₡ = { 50,    },
    † = {100,100},
    ‡ = { 80, 80},
    • = {    ,100},
    · = {300,400}, % periodcentered
   /twodotenleader = {150,200},
   /ellipsis   = {100,150},
    ℃ = { 80,    },
    ° = {400,400},
    ™ = {100,100},
    © = {100,100},
    ® = {100,100},
    ª = {200,200},
    º = {200,200},
    ¹ = {200,200},
    ² = {200,200},
    ³ = {200,200},
    ¬ = {200,    },
    ¶ = {    ,100},
    − = {300,300}, % minus
    ± = {150,200},
    × = {100,150},
    ÷ = {150,200},
    € = { 50,100},
    ¥ = { 50, 50},
  % Greek
    Γ = {    ,150},
    Δ = {100,100},
    Θ = { 50, 50},
    Λ = {100,100},
    Ξ = { 50, 50},
    ϒ = {100,100},
    Φ = { 50, 50},
    Ψ = { 50, 50},
    Ω = {    , 50},
    ζ = {    , 50},
    λ = { 50, 50},
    γ = { 50, 50},
    π = { 50, 50},
    ρ = {    , 50},
    σ = { 50, 50},
    τ = { 50, 50},
    χ = { 50, 50},
    φ = { 50, 50},
    ϙ = { 50, 50},
    ψ = { 50, 50},
  % Cyrillic
    Г = {    ,150},
    Д = { 50, 50},
    Ж = { 50, 50},
    К = {    , 50},
    Л = { 50,    },
    Љ = { 50, 50},
    З = { 50, 50},
    У = { 50,100},
    Ф = { 50, 50},
    Ч = { 70,    },
    Я = { 50,    },
    Ъ = { 50, 50},
    Ь = {    , 50},
    ж = { 50, 50},
    ф = { 50, 50},
    ъ = { 50, 50},
    Ѱ = { 50, 50},
    r = {    , 50},
    V = { 50, 50},
  % other
    Þ = {    , 50},
    þ = {    , 50},
    Λ = {100,100},
    (ı)  = { 35, 65},
    (a)  = { 30, 60},
  }

\SetProtrusion
  [ name     = EBGaramond-it-TU,
    load     = EBGaramond-it-T1-LF ]
  { encoding = {TU,EU1,EU2},
    family   = EBGaramond,
    shape    = it }
  {
   /zero.tosf  = {150,150},
   /one.tosf   = {150,150},
   /two.tosf   = {80,80},
   /three.tosf = {50,80},
   /four.tosf  = {50,80},
   /five.tosf  = {50,80},
   /six.tosf   = {50,50},
   /seven.tosf = {50,100},
   /eight.tosf = {50,50},
   /nine.tosf  = {50,80},
   /one.lf     = {50,50},
   /two.lf     = {50,50},
   /three.lf   = {80,50},
   /four.lf    = {50,50},
   /five.lf    = {50,50},
   /six.lf     = {50,50},
   /seven.lf   = {50,50},
   /eight.lf   = {50,50},
   /nine.lf    = {50,  },
   /one.osf    = {50,50},
   /two.osf    = {50,50},
   /three.osf  = {  ,80},
   /four.osf   = {50,50},
   /seven.osf  = {50,50},
  % textcomp
    \textquotesingle = {800,100},
    − = {300,300}, % minus
    ˷ = {200,250},
    † = {200,100},
    ‡ = { 80, 80},
    • = {300,    },
    ℃ = {200,    },
    ƒ = {100,    },
    ₡ = {100,    },
    ™ = {200,    },
    © = {200,100},
    ® = {200,100},
    ¬ = {300,    },
    ° = {500,100},
    ± = {200,150},
    ¹ = {300,100},
    ² = {300,    },
    ³ = {300,    },
    · = {300,500}, % periodcentered
   /twodotenleader = {150,300},
   /ellipsis   = {100,200},
    € = {100,    },
    × = {200,100},
    ÷ = {200,200},
    ¶ = {    ,100},
    ª = {200,200},
    º = {200,200},
    ¥ = { 50, 50},
  % Greek
    Δ = {150,    },
    Θ = { 50,    },
    Λ = {150,    },
    ϒ = {100, 50},
    Φ = { 50,    },
    Χ = { 50,    },
    Ψ = {100,    },
    Ω = { 50,    },
    γ = {    , 50},
    λ = { 50,    },
  % Cyrillic
    Ү = { 50,    },
    Ч = {100,    },
    З = {100,    },
  % other
    Þ = { 50, 50},
    þ = {    , 50},
  }

\SetProtrusion
  [ name     = EBGaramond-sc-TU,
    load     = EBGaramond-TU ]
  { encoding = {TU,EU1,EU2},
    family   = EBGaramond,
    shape    = sc }
  {
     a = {50,50},
   \ae = {50,  },
     d = {  ,50},
     f = {  ,50},
     g = {50,  },
     j = {50,  },
     l = {  ,50},
     o = {50,50},
   \oe = {50,  },
     q = {50,70},
     r = {  , 0},
     t = {50,50},
     y = {50,50},
  % Greek
     α = {50,50},
     γ = {  ,50},
     δ = {50,50},
     λ = {50,50},
     ο = {50,50},
     τ = {50,50},
     υ = {50,50},
     ψ = {50,50},
  % Cyrillic
     т = {50,50},
  }

\SetProtrusion
  [ name     = EBGaramond-scit-TU,
    load     = EBGaramond-it-TU ]
  { encoding = {TU,EU1,EU2},
    family   = EBGaramond,
    shape    = scit }
  {
     a = {50,50},
   \ae = {50,  },
     d = {  ,50},
     f = {  ,50},
     g = {50,  },
     j = {50,  },
     l = {  ,50},
     o = {50,50},
   \oe = {50,  },
     q = {50,70},
     r = {  , 0},
     t = {50,50},
     y = {50,50},
  % Greek
     α = {50,50},
     γ = {  ,50},
     δ = {50,50},
     λ = {50,50},
     ο = {50,50},
     τ = {50,50},
     υ = {50,50},
     ψ = {50,50},
  % Cyrillic
     т = {50,50},
  }
⟨/EBGaramond⟩
```

3.2.4 Palatino

```
⟨*Palatino⟩
\SetProtrusion
   [ name     = palatino-default ]
   { encoding = {TU,EU1,EU2},
     family   = {Palatino} }
   {
     A = {50,50},
     D = {  ,50},
     J = {50,  },
     K = {  ,50},
     L = {  ,50},
     O = {25,  },
     T = {50,50},
     V = {50,50},
     W = {50,50},
     X = {50,50},
     Y = {50,50},
     b = {  ,25},
     d = {25,30},
     f = {  ,50},
     g = {  ,100},
     k = {  ,50},
     p = {  ,50},
     q = {50,  },
     r = {  ,50},
     t = {  ,50}, � = { ,50}, � = {  ,50},
     v = {75,50},
     w = {50,50},
     x = {50,50},
     y = {50,70},
     1 = {100,50},
     2 = {25,50},
     4 = {50,  },
     6 = {50,  },
     9 = {25,  },
        Æ  = {100,  },
        Œ  = {25,   },
        .  = {  ,700},          ‥  = {  ,350},      …  = {,150},
       {,}= {  ,500},
        :  = {  ,500},
        ;  = {  ,500},
        !  = {  ,100},          ‼  = {  ,100},
        ?  = {  ,200},          ‽  = {  ,200},
        @  = {50,50},
        ~  = {200,250},
        &  = {50,100},
       \%  = {100,100},
        *  = {200,200},
        +  = {250,250},
        (  = {100,    },        )  = {    ,300},
        /  = {200,300},
        -  = {400,500},
        \textendash          = {300,300},    \textemdash           = {200,200},
        \textquoteleft       = {500,700},    \textquoteright       = {500,700},
        \textquotedblleft    = {300,400},    \textquotedblright    = {300,400},
        \textbackslash       = {200,300},
        \quotesinglbase      = {400,400},    \quotedblbase         = {400,400},
        \guilsinglleft       = {400,400},    \guilsinglright       = {300,500},
        \guillemotleft       = {300,300},    \guillemotright       = {200,400},
        \textexclamdown      = {100,    },   \textquestiondown     = {100,    },
        \textbraceleft       = {400,200},    \textbraceright       = {200,400},
        \textless            = {200,100},    \textgreater          = {100,200},
        ≤                    = {200,100},    ≥                     = {100,200},
        \textminus                  = {300,300},
        \texttrademark              = {200,200},
        \textcopyright              = {200,200},
        \textregistered             = {200,200},
        \textdegree                 = {300,300},
        ¦                    = {450,500},    ¬                     = {250,150},
        ⌐                    = {150,250},
        ·                           = {850, 700},
        ¶                           = {100,0},
        ×                           = {150, 300},
        ª                    = {300,300},    º                     = {300,300},
        ⁰  = {200,400},
        ¹  = {400,350},          ²  = {200,300},          ³  = {250,400},
        ⁴  = {250,350},          ⁵  = {200,300},          ⁶  = {250,400},
        ⁷  = {200,450},          ⁸  = {250,400},          ⁹  = {200,350},
        ₀  = {200,400},
        ₁  = {400,250},          ₂  = {200,300},          ₃  = {250,400},
        ₄  = {250,350},          ₅  = {200,300},          ₆  = {250,400},
        ₇  = {200,450},          ₈  = {250,400},          ₉  = {200,350},
        ±  = {150,100},                                   ÷  = {300,300},
        þ  = {   ,25},
        ₊  = {300,450},          ₋  = {300,450},
        ⁺  = {300,450},          ⁻  = {300,450},
        †                    = {200,250},    ‡                     = {200,250},
        π  = {50,   },
        ſ  = {   ,50},
        №  = {100,150},
        \textservicemark            = {100,200},
        -  = {400,500},          ‐  = {400,500},          −  = {200,300},
        –  = {205,305},          —  = {200,300},          ―  = {50,150},
        •  = {125,200},
%          /a.sc  = {50,50},

    }

\SetProtrusion
    [ name     = palatino-it    ]
    { encoding = {TU,EU1,EU2},
        family   = {Palatino},
        shape    = {it,sl}  }
    {
        A  = {50,50},
        Æ  = {50,   },
        B  = {50,   },
        C  = {50,   },
        D  = {50,50},
        E  = {50,   },
        F  = {50,   },
        G  = {50,   },
        H  = {50,   },
        K  = {50,   },
        L  = {50,   },
        O  = {50,   },
        Œ  = {50,   },
        P  = {50,   },
        Q  = {50,   },
        R  = {50,   },
        S  = {50,   },
        $  = {50,   },
        T  = {100, },
        U  = {50,   },
        V  = {100,50},
        W  = {50,   },
        X  = {50,   },
        Y  = {100,50},
        b  = {   ,50},
        c  = {25,   },
        g  = {75,   },
        i  = {25,   },
        m  = {   ,50},
        n  = {   ,50},
        p  = {   ,25},
        q  = {25,   },
        x  = {   ,50},
        1  = {100, },
        2  = {50,   },
        4  = {50,   },
        7  = {50,   },
        .  = {  ,500},          ‥  = {  ,350},      …  = {  ,200},
       {,}= {  ,500},
        :  = {  ,300},
        ;  = {  ,300},
        ?  = {  ,300},          ‽  = {  ,300},
        &  = {50,50},
       \%  = {100,100},
        *  = {200,200},
        +  = {150,200},
        @  = {50,50},
        ~  = {200,150},
        (  = {200, },       )  = {   ,200},
        /  = {100,200},
        -  = {300,500},
        \textendash          = {300,300},    \textemdash           = {200,200},
        \textquoteleft       = {700,400},    \textquoteright       = {700,400},
        \textquotedblleft    = {500,300},    \textquotedblright    = {500,300},
        _  = {100,100},
        \textbackslash       = {100,200},
        \quotesinglbase      = {500,500},    \quotedblbase         = {400,400},
        \guilsinglleft       = {400,400},    \guilsinglright       = {300,500},
        \guillemotleft       = {300,300},    \guillemotright       = {300,300},
        \textexclamdown      = {100,     },  \textquestiondown     = {200,     },
        \textbraceleft       = {200,100},    \textbraceright       = {200,200},
        \textless            = {300,100},    \textgreater          = {200,100},
        ≤                    = {200,100},    ≥                   = {100,200},
        ¦                    = {450,500},    ¬                   = {250,150},
        ·                             = {850, 700},
        ¶                              = {100,0},
        ×                              = {150, 300},
        ª = {300,250},          ° = {300,300},          º = {300,250},
        ⁰ = {300,200},
        ¹ = {300,150},          ² = {350,200},          ³ = {250,150},
        ⁴ = {350,100},          ⁵ = {300, 50},          ⁶ = {400,100},
        ⁷ = {400, 50},          ⁸ = {250, 50},          ⁹ = {300, 50},
        ₀ = {300,300},
        ₁ = {300,350},          ₂ = {300,150},          ₃ = {250,250},
        ₄ = {400,200},          ₅ = {300,100},          ₆ = {450,200},
        ₇ = {450,150},          ₈ = {400,250},          ₉ = {400,200},
        ± = {150,100},                                   ÷ = {300,300},
        þ = { 50,    },
        †                   = {250,200},     ‡                  = {250,200},
        ₊ = {300,450},          ₋ = {300,450},
        ⁺ = {300,450},          ⁻ = {300,450},
        - = {300,500},          ‑ = {300,500},          − = {100,300},
        – = {125,305},          — = {200,300},          ― = {125,150},
        • = {125,200}
      }

    \SetProtrusion
      [ name     = palatino-sc,
        load     = palatino-default ]
      { encoding = {TU,EU1,EU2},
        family   = {Palatino},
        shape    = sc }
      {
        a = {50,50},
        æ = {50,  },
        b = { 0, 0},
        d = { 0, 0},
        f = { 0, 0},
        g = { 0, 0},
        j = {50,  },
        l = {  ,50},
        o = { 0, 0},
        p = { 0, 0},
        q = { 0,  },
        r = {  , 0},
        t = {50,50},
        y = {50,50},
        fl = { 0,50},
        ffl = { 0,50},
        � = { 0,50},
        � = { 0,50}
      }
    ⟨/Palatino⟩
```

3.2.5 Basic glyph set

The protrusion settings will still be loaded from microtype.cfg.

```
⟨TU-basic⟩ %% No settings.
```

3.2.6 Empty glyph set

```
⟨*TU-empty⟩
\SetProtrusion
  [ name = empty ]
  { encoding = {TU,EU1,EU2},
    family   = {TU-empty} }
  { }
⟨/TU-empty⟩

```

4 Auxiliary file for micro fine tuning

This file may be used to test protrusion and (less so) expansion settings.

```
⟨*test⟩
\documentclass{article}
%% options are passed through to microtype
\usepackage[stretch=50]{microtype-show}

%% options for microtype-show
\ShowGlyphIndextrue
\ShowMissingGlyphstrue
\def\GlyphScaleFactor{2}

%% load any required font packages:
\ifpdftex
 \usepackage[T1]{fontenc}
\else
 \usepackage{fontspec}
\fi

\begin{document}
\microtypesetup{expansion=false}

%% load your font here:

\ShowCharacterInheritance

\newpage
\ShowProtrusion

\newpage
%% show single glyphs
 %\ShowDummyLine
 %\ShowProtrusionLineGlyph{A}
 %\ShowProtrusionLineIndex{27}

%% loop through all glyphs of the font;
%% protrusion values are shown in 1000th of 1em
 \ShowProtrusionDefined

 %\ShowProtrusionMissing

 %\ShowProtrusionAll

\newpage %% ------------------------------------------------------------------
This is the current font stretched by 5\%, normal, and shrunk by 5\%:

\newlength{\MTln}
\newcommand*\teststring
  {ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz}
\settowidth{\MTln}{\teststring}
\microtypesetup{expansion=true}

\bigskip\noindent\parbox{1.05\MTln}{\teststring\linebreak\\\teststring}\par
\bigskip\noindent\parbox{0.95\MTln}{\teststring}
\end{document}
⟨/test⟩
```

Needless to say that things may always be improved. For suggestions, mail to w.m.l@gmx.net or file an issue at https://github.com/schlcht/microtype/issues.

A The title logo

This is microtype-logo.dtx. You may treat this file in three different ways:

- compile it by itself
- \input it in the body of a dtx file
- \input it in the preamble: it then provides the command \printlogo, which will do just that

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

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

```
⟨*logo⟩
```

Here's how the logo on the title page was created.[23] 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.[24] It will show:

- the character
- the TEX box
- the bounding box
- kerns

A.1 Macros

To run this file, TEX needs to find the afm file (either in the TEXINPUTS path, or in the current working directory).

First input fontinst.

```
\input fontinst.sty
```

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

```
\input bbox.sty
```

\tempdim Allocate some dimen registers.

```
\newdimen\tempdim
```

\fboxrulei Frame width of the box as TEX sees it.

```
\newdimen\fboxrulei
\fboxrulei=0.1pt
```

\fboxruleii Frame width of the bounding box.

```
\newdimen\fboxruleii
\fboxruleii=0.1pt
```

\kernboxheight Height of the box indicating the kern.

```
\newdimen\kernboxheight
\kernboxheight=5pt
```

\scaletoem An auxiliary macro. Return a dimension relative to the em-width of the font. Requires e-TEX.

```
\setcommand\scaletoem#1{\dimexpr #1 sp*\fontdimen6\font/1000\relax}
```

\showlogo A fontinst incantation whose sole purpose is to produce the logo. Its argument is a string (letters only).

```
\fontinstcc
\def\showlogo#1{%
```

Some fonts do not specify the \fontdimen 6 (width of an em) in the afm file. In this case, use the font size, which is correct in most cases.

```
   \ifdim\fontdimen6\font = 0pt
      \typeout{***~Warning:~no~fontdimen~6~specified~***^^J%
               ***~setting~it~to~\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi~***}
      \fontdimen6\font=\pdffontsize\font \ifnum\pdftexversion < 130 pt\fi\relax
   \fi
  \installfonts
    \input_metrics{}{\logofont,\metrics\printbbs{#1}\relax}
```

[23] 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.

[24] Message ID: 42aa3687$0$24366$9b4e6d93@newsread2.arcor-online.net

```
  \endinstallfonts
}
\normalcc
```

Layers.

```
\makeatletter
\def\mtl@layer#1#2{\pdfliteral{/OC/#1 BDC}#2\pdfliteral{EMC}}
\ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
\ifx\mt@order  \@undefined\let\mt@order  \@empty\fi
\xdef\mt@order{\mt@order[(Logo)}
\let\mtl@resources\@empty
\def\mtl@register#1{%
  \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
  \expandafter\xdef\csname mtl@#1\endcsname{\the\pdflastobj\space 0 R }
  \xdef\mt@objects{\mt@objects\csname mtl@#1\endcsname}
  \xdef\mt@order{\mt@order\csname mtl@#1\endcsname}
  \xdef\mtl@resources{\mtl@resources/#1 \csname mtl@#1\endcsname}}
\mtl@register{canvas}
\mtl@register{characters}
\mtl@register{bounding-boxes}
\mtl@register{TeX-boxes}
\xdef\mt@order{\mt@order]}
\global\let\mtl@objects\mt@objects
\def\togglelayer#1#2{%
  \pdfstartlink width \wd\logobox height \ht\logobox depth \dp\logobox
    user{/Subtype/Link
         /BS << /Type/Border/W 0 >> /H/O
         /A << /S/SetOCGState
               /State[/Toggle \csname mtl@#1\endcsname] >>
  }#2\pdfendlink
}
```

\printbbs Preparation.

```
\setcommand\printbbs#1{%
  \setbox0\hbox{#1}%
  \leavevmode
  \kern-\fboxrulei
```

The canvas in the natural width of the text minus protrusion, in color bgcolor.

```
  \mtl@layer{canvas}{%
    \getboundarychars#1\relax
    \tempdim=\dimexpr\wd0 - (\scaletoem{\lpcode\font\firstchar}+
                             \scaletoem{\rpcode\font\lastchar})\relax
    \kern\dimexpr\scaletoem{\lpcode\font\firstchar}\relax
    \lower\dimexpr\dp0+0.05em \relax \vbox{\color{bgcolor}%
          \hrule width  \tempdim
                 height \dimexpr\dp0+\ht0+0.15em\relax}%
    \kern-\tempdim
```

The baseline, in color blcolor.

```
    \vbox{\color{blcolor}%
          \hrule width  \tempdim
                 height \fboxrulei}%
  }%
  \kern-\dimexpr\wd0 -\scaletoem{\rpcode\font\lastchar}\relax
```

The string.

```
  \printbbss #1\relax\relax
}
```

\getboundarychars Get first . . . .

```
\def\getboundarychars#1#2\relax{%
   \def\firstchar{`#1}%
   \getlastchar#1#2\relax
}
```

\getlastchar . . . and last character.

```
\def\getlastchar#1#2{%
```

```
  \ifx\relax#2\relax
     \def\lastchar{`#1}%
  \else
     \expandafter\getlastchar
  \fi #2%
}
```

\printbbss Loop over all characters of the string.

```
\def\printbbss#1#2#3\relax{%
  \ifx\relax#1\relax
  \else
     \ifx\relax#2\relax
        \printbb{#1}{}%
     \else
        \printbb{#1}{#2}%
     \fi
     \expandafter\printbbss
  \fi #2#3\relax
}
```

\printbb Record the kern between the current and the following character, then print the character. `\kerning` is a `fontinst` command.

```
\setcommand\printbb#1#2{%
  \setbox0\hbox{\kerning{#1}{#2}\xdef\thekern{\number\result}}%
  \showboxes{#1}%
```

This could be another application.

```
%       \quad
%     w: \the\scaletoem{\width{#1}},
%     bb: \the\scaletoem{\bbleft{#1}}/%
%         \the\scaletoem{\bbright{#1}},
%         \the\scaletoem{\number\numexpr\width{#1}-\bbright{#1}\relax}
%     h: \height{#1}/\bbtop{#1}, \bbbottom{#1}/\depth{#1}\par
}
```

\showboxes Print the boxes for char ⟨*#1*⟩. This won't work if ⟨*#1*⟩ isn't also the PostScript name of the glyph (e.g., 'comma' ≠ ',').

```
\setcommand\showboxes#1{%
 \leavevmode
 \color{texcolor}%
```

We have to record the width of the glyph.

```
 \setbox0\hbox{{\color{textcolor}#1}}%
 \global\tempdim=\wd0\relax
 \kern-\fboxrulei
```

1. *The TeX box:* Print a frame in color `texcolor`. This frame shows the glyph as TeX sees it.

```
    \mtl@layer{TeX-boxes}{%
      \hbox{%
        \lower\dimexpr \dp0 + \fboxrulei\relax
        \hbox{%
          \vbox{%
            \hrule height\fboxrulei
            \hbox{%
              \vrule width\fboxrulei height \dimexpr\ht0 + 2\fboxrulei\relax
              \phantom{\unhcopy0}%
              \vrule width\fboxrulei
            }%
            \hrule height\fboxrulei}}}%
    }%
```

2. *The character:* Now we step back and print the actual glyph. We hold it back until now, so that it will be printed *on top* of its box.

```
    \kern-\wd0
    \mtl@layer{characters}{\hbox{\box0}}%
```

Step back by the amount that the character's bounding box differs from the TeX box on the left side.

```
    \kern\dimexpr\scaletoem{\bbleft{#1}}-\tempdim-\fboxruleii\relax
```

3. *The bounding box:* will be printed in color `bbcolor`.

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

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

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

`\printlogo`

```
\newbox\logobox
\def\printlogo{%
  \setbox\logobox=\hbox{\vbox{%
    \MakePercentComment
```

This is the Kepler MM font used in the logo.

```
    \def\logofont{pkpri9e10}
    \transformfont{\logofont}{\reencodefont{8r}{\fromafm{pkpmmri8a10}}}
    \font\thelogofont=\logofont\space at 82pt
```

This would load the italic Palatino font instead.

```
%\def\logofont{pplri}
%\transformfont{\logofont8r}{\reencodefont{8r}{\fromafm{\logofont8a}}}
%\edef\logofont{\logofont8r}
%\font\thelogofont=\logofont\space at 78pt
```

Load the font.

```
    \thelogofont
```

Protrusion values (overdone for didactic reasons).

```
    \lpcode\font`M=96
    \rpcode\font`e=46
```

Now we can generate the logo.

```
    \pdfliteral direct{/SXS gs}%
    \showlogo{Microtype}%
%    \rlap{\normalfont\normalsize\raisebox{55pt}{\footnotemark[1]}}%
%    \kern5pt\\[3\baselineskip]
%  \long\def\@makefntext##1{%
%    \leftskip 0pt
%    \parindent 0pt
%    \everypar{\parindent 0pt}%
%    \leavevmode\hbox to 15pt{\@thefnmark\hss}##1}
%  \footnotetext[1]{This graphic displays on a
%    \togglelayer{canvas}{canvas} the \togglelayer{characters}{characters},
%    their \togglelayer{bounding-boxes}{bounding boxes}
%    and \togglelayer{TeX-boxes}{\TeX\ boxes}.}
  }}%
  \edef\logodimens{width \the\wd\logobox height \the\ht\logobox depth \the\dp\logobox}
  \immediate\pdfobj{<</Type/ExtGState /CA 0.6 /ca 0.6 /BM/Normal >>}%
  \immediate\pdfxform
            attr {/Group <</Type/Group /S/Transparency /I true /CS/DeviceRGB >>}
            resources {/Properties <<\mtl@resources>>
                       /ExtGState << /SXS \the\pdflastobj\space 0 R >> }
            \logobox
%  \vskip-2.5\baselineskip
%  \leavevmode
%  \togglelayer{characters}{%
%    \pdfrefxform\pdflastxform
%  }%
   \pdfannot\logodimens{%
       /Subtype/Widget /FT/Btn /T(Logo)
       %/F 4 % why did I say this?
       /AP << /N \the\pdflastxform\space 0 R >>
       /AA << /E << /S/SetOCGState /State[/Toggle \mtl@characters] >>
              /X << /S/SetOCGState /State[/Toggle \mtl@characters] >>
              /D << /S/SetOCGState /State[/Toggle \csname mtl@bounding-boxes\endcsname] >>
              /U << /S/SetOCGState /State[/Toggle \csname mtl@TeX-boxes\endcsname] >>
           >> }%
  \vspace{3\baselineskip}
}
\IfFileExists{pkpmmri8a10.afm}\relax{\def\printlogo{\MT@warning{File pkpmmri8a10.afm not found.
  \MessageBreak Cannot create logo}}}
```

Our font.

```
\pdfmapline{+pkpmmri8r10 KeplMM-It_385_575_10_ " TeXBase1Encoding ReEncodeFont " <8r.enc <pkpmmri8a10.pfb}
```

Define colours (thered and thegreen are copied from microtype.dtx).

```
\def\mtdefinecolors{
 \definecolor{thered}{rgb}{0.65,0.04,0.07}
 \definecolor{thegreen}{rgb}{0.06,0.44,0.08}
 \colorlet{texcolor}{thegreen!50}  % TeX boxes
 \colorlet{kerncolor}{texcolor}    % negative kerns
 \colorlet{bbcolor}{thered!50}     % bounding box
 \colorlet{bgcolor}{black!8}       % canvas
 \colorlet{blcolor}{black!50}      % baseline
 \colorlet{textcolor}{black!40}    % text
}
```

Use with microtype.dtx

```
\ifx\documentclass\@twoclasseserror
  \usepackage[xcdraw]{xcolor}
  \mtdefinecolors
```

```
\else
```

A.2 Document

Now we can start the document.

```
\documentclass[10pt,a4paper]{ltxdoc}
\providecommand\MakePercentComment{\relax}
\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
```

Re-use the preamble from `microtype.dtx`.

```
\usepackage{microtype-doc}
\usepackage{attachfile}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects] /D << /Order [\mt@order] >> >>}
\makeatother
\begin{document}
```

You are currently reading this.

```
\DocInput{microtype-logo.dtx}
\newpage
And here it is:\vspace{6\baselineskip}
\begin{center}
  \printlogo
\end{center}
\expandafter\enddocument
\fi
```

That's it.

```
⟨/logo⟩
```

B The letterspacing illustration

This is `microtype-lssample.dtx`. You may treat this file in three different ways:

- compile it by itself
- `\input` it in the body of a `dtx` file
- `\input` it in the preamble: it then provides the commands
  - `\lssample`: prints the letterspacing illustration
  - `\anchorarrow`: anchors an arrow for layer ⟨*#1*⟩
  - `\showarrow`: toggles layer ⟨*#1*⟩ or ⟨*#2*⟩, and prints ⟨*#2*⟩

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

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

```
\ifx\lssample\undefined
⟨*lssample⟩
```

Upon popular request, here's how I've created the letterspacing illustration.[25]

B.1 Macros

Rule width and image height and depth.

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

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

Our font (Adobe Caslon).

```
\def\lsfont{\fontfamily{paca}\selectfont}
```

Loop over all letters in ⟨*#2*⟩, letterspacing them by ⟨*#1*⟩.

```
\def\dols#1#2{\lsamount=#1\relax \dolss#2\enddols}
\def\dolss#1#2\enddols{%
  \ifx\empty#2\empty\divide\lsamount 2\fi
  \ls{#1}%
  \ifx\empty#2\empty\else \dolss#2\enddols \fi
}
```

One tikz picture for each letter.

```
\def\ls#1{%
  \begin{tikzpicture}[remember picture,line width=\lsrule]
    \tikzstyle{every node}=[inner sep=0pt]
```

The bounding box.

```
    \mts@layer{stuff}{%
      \node[draw=thegrey,
            fill=theshade,
            outer sep=\lsrule,
            anchor=base,
            font=\lsfont]{\phantom{#1}};
    }
```

The letter.

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

Two auxiliary coordinates.

```
    \path (#1.south west) ++(+.5\lsrule,-.5\lsrule) coordinate (#1L);
    \path (#1.base east)  ++(-.5\lsrule,-\lsdepth)  coordinate (#1R);
    \mts@layer{stuff}{%
```

Now draw the normal character width,

```
      \draw[color=thered!75,
            fill=thered!30,
            outer sep=\lsrule]
            (#1L) rectangle (#1R);
      \ifdim\lsamount>0pt
        \path (#1.base east) ++(+.5\lsamount,-6pt) coordinate (#1_ls);
        \path (#1R) ++(\lsamount+\lsrule,+\lsdepth) coordinate (#1E);
```

and the letter space.

```
        \draw[color=thered,
              fill=thered!50,
              outer sep=\lsrule]
              (#1R) ++(+\lsrule,+0pt) rectangle (#1E);
      \fi
    }
  \end{tikzpicture}%
  \ignorespaces
}
```

Draw the interword space.

```
\def\lssp#1#2#3#4{%
  \begin{tikzpicture}[remember picture,line width=\lsrule,inner sep=0pt]
    \mts@layer{stuff}{%
      \tikzstyle{every draw}=[anchor=bottom]
      \coordinate(#1space) at (#2/2,\lsdepth/2);
      \coordinate(#1stretch) at (#2+#3/2,+0pt);
      \coordinate(#1shrink) at (#2-#4/2,+0pt);
      \draw[color=thegreen,fill=thegreen!50,use as bounding box]
            (0,0) rectangle ++(+#2,+\lsdepth);
      \draw[color=thegreen,fill=thegreen!30]
            (+#2,-\lsrule) rectangle ++(+#3,-4pt+\lsrule);
      \draw[color=thegreen,fill=thegreen!50]
            (+#2,-\lsrule) rectangle ++(-#4,-4pt+\lsrule);
      \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!50]
```

```
            (+#2,-2pt-.5\lsrule) -- ++(+#3,+0pt);
      \draw[->,line width=0.3pt,shorten <=0.5\lsrule,color=thegreen!30]
            (+#2,-2pt-.5\lsrule) -- ++(-#4,+0pt);
    }%
  \end{tikzpicture}%
  \ignorespaces
}
```

Layers.

```
\def\mts@layer#1#2{\pdfliteral page{/OC/#1 BDC}#2\pdfliteral page{EMC}}
\def\mtsx@layer#1#2{\pdfliteral page{/OC/stuff BDC /OC/#1 BDC}#2\pdfliteral page{EMC EMC}}
\ifx\mt@objects\@undefined\let\mt@objects\@empty\fi
\ifx\mt@order  \@undefined\let\mt@order  \@empty\fi
\xdef\mt@order{\mt@order[(Sheep)}
\let\mts@resources\@empty
\def\mts@register#1{%
  \immediate\pdfobj{<< /Type/OCG /Name(#1) >>}
  \expandafter\xdef\csname mts@#1\endcsname{\the\pdflastobj\space 0 R }
  \xdef\mt@objects{\mt@objects\csname mts@#1\endcsname}
  \xdef\mt@order{\mt@order\csname mts@#1\endcsname}
  \xdef\mts@resources{\mts@resources/#1 \csname mts@#1\endcsname}}
\mts@register{stuff}
\mts@register{tracking}
\mts@register{ispace}
\mts@register{ospace}
\mts@register{istretch}
\mts@register{ishrink}
\mts@register{ostretch}
\mts@register{oshrink}
\mts@register{okern}
\mts@register{ligature}
\mts@register{_compatibility}
\xdef\mt@order{\mt@order]}
```

Anchor point for the arrow in the code.

```
\newcommand\anchorarrow[1]{%
  \tikz[remember picture,overlay]\node(#1_c){};}
```

Add an arrow from code to image.

```
\newcommand\add@arrow[5][left]{%
  \tikz[remember picture,overlay,bend angle=14,looseness=0.75,>=latex]{%
    \mtsx@layer{#3}{\draw[->,thick,color=the#2](#4) to[bend #1] (#5);}}%
}
```

Toggle layer.

```
\def\toggle@layer#1#2#3{%
  \pdfstartlink
    user{/Subtype/Link
         /BS << /Type/Border/W 0 >> /H/O
%         /BS << /Type/Border/W 1 /S/D /D[4 1] >>
%         /C[0.7 0.7 0.7] /H/O
         /Contents(Click to Toggle!)
         /A << /S/SetOCGState
               /State[/Toggle \csname mts@#1\endcsname] >> }%
  \rlap{#2}%
  {\fboxsep=0pt \fboxrule=0pt
   \mtsx@layer{stuff}{%
     \rlap{\fcolorbox{white}{white}{\vphantom{kg}\color{the#3}#2}}}%
   \mtsx@layer{#1}{%
     \fcolorbox{white}{the#3!50}{\vphantom{kg}\color{white}#2}}%
  }%
  \pdfendlink
}
\newcommand\showarrow[2][]{%
  \ifx\relax#1\relax\def\@tempa{#2}\else\def\@tempa{#1}\fi
  \toggle@layer{\@tempa}{{\itshape #2}}}
```

The environment for our illustration.

```
\def\ls@sample#1{{%
  \parskip  4pt \parindent 0pt
  \par
  \vskip4pt
  {\leftskip 15pt
   \mt@pseudo@marg{\color{theblue}Click on the image to show the kerns
      and spacings involved. Click on emphasised words in the text below
      to reveal the relation of image and code.\strut}
   \mt@layer{_compatibility}{%
      \mt@place{\rlap{\hskip-\marginparwidth \color{white}%
        \vrule width\dimexpr\hsize+\marginparwidth\relax height\mt@unvdimen}}
      \mt@pseudo@marg{\color{thered}%
        If you had a \acronym{PDF} viewer that understands
        \acronym{PDF}\,{\smaller1.5}, you could hide the arrows selectively.}}
   \vskip-\mt@unvdimen}%
  \vskip-4pt
  \setlength\fboxsep{4pt}%
  \leavevmode
  \pdfstartlink
    user{/Subtype/Link
         /BS << /Type/Border/W 0 >> /H/O
         /A << /S/SetOCGState
               /State[/Toggle \mts@stuff] >> }%
    \fcolorbox{theframe}{theshade}%
      {\fontsize{34}{38}\selectfont #1}%
  \pdfendlink
  \par\medskip
  }%
  \edef\x{\pdfpageresources{/Properties <<\mts@resources>>}}\x
}
```

Now define the illustration to be used in the document.

```
\def\lssample{%
  \ls@sample{%
    \dols{0pt}{Stop}
      \lssp{o}{0.45em}{0.25em}{0.15em}
    \dols{0.16em}{{st}ealing}\hskip-\dimexpr 0.08em+\lsrule\relax
      \lssp{i}{13.82pt}{4.65pt}{2.08pt}
    \dols{0.16em}{sheep}
    \dols{0pt}{!}
  }%
```

Don’t forget to add the arrows.

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

This is for use with microtype.dtx

```
\ifx\documentclass\@twoclasseserror
  \usepackage{tikz}
\else
```

B.2 Document

```
\documentclass[10pt,a4paper]{ltxdoc}
\expandafter\def\csname ver@microtype.dtx\endcsname{2999/99/99}
```

Re-use the preamble from microtype.dtx.

```
\usepackage{microtype-doc}
\usepackage{attachfile}
\usepackage{tikz}
\makeatletter
\pdfcatalog{/OCProperties << /OCGs [\mt@objects]
                             /D << /Order [\mt@order] /BaseState/OFF >> >> }
\makeatother
\begin{document}
```

You are currently reading this.

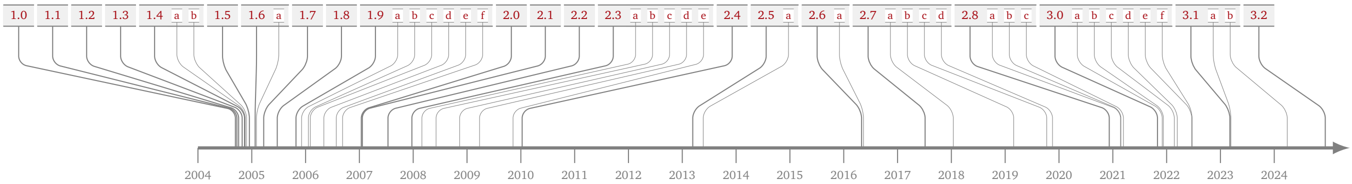
```
\DocInput{microtype-lssample.dtx}
```

Now show what we are able to do.

```
\noindent
 Since a picture is worth a thousand words, probably even more if, in our
 case, it depicts a couple of letterspaced words, let's bring one to sum up
 these somewhat confusing options. Suppose you had the following settings
 (which I would in no way recommend; they are only for illustrative purposes):
\begin{verbatim}
\SetTracking
  [ no ligatures  = {"\anchorarrow{nolig}"f},
    spacing       = {60"\anchorarrow{ispace}"0*,"%
                      "-1"\anchorarrow{istretch}"00*, "\anchorarrow{ishrink}"},
    outer spacing = {4"\anchorarrow{ospace}"50,"%
                      "2"\anchorarrow{ostretch}"50,1"\anchorarrow{oshrink}"50},
    outer kerning = {"\anchorarrow{okernbegin}"*,"%
                      \anchorarrow{okernend}"*} ]
  { encoding = * }
  { 1"\anchorarrow{lsamount}"60 }
\end{verbatim}
 and then write:
\begin{verbatim}
Stop \textls{stealing sheep}!
\end{verbatim}
 this is the (typographically dubious) outcome:

 \lssample

\noindent
 While the word `Stop' is not letterspaced, the space between the letters in
 the other two words is expanded by the \showarrow[tracking]{tracking~amount}{red}
 of 160/1000\,em\,=\allowbreak\,0.16\,em.
 The \showarrow[ispace]{inner~space}{green} within the letterspaced text is
 increased by 60\%, while its \showarrow[istretch]{stretch}{green} amount is
 decreased by 10\% and the \showarrow[ishrink]{shrink}{green} amount is left
 untouched.
 The \showarrow[ospace]{outer~space}{green} (of 0.45\,em) immediately before the
 piece of text may \showarrow[ostretch]{stretch}{green} by 0.25\,em and
 \showarrow[oshrink]{shrink}{green} by 0.15\,em.
 Note that there is no outer space after the text, since the exclamation mark
 immediately follows; instead, the default \showarrow[okern]{outer~kern}{red}
 of half the letterspace amount (0.08\,em) is added.
 Furthermore, one \showarrow{ligature}{grey} wasn't broken up, because we
 neglected to specify the `|s|' in the |no ligatures| key.

\expandafter\enddocument
\fi
```

⟨/lssample⟩

C Change history



Numbers prefixed with ‘U’ refer to the User manual.

2004/09/11 Version 1.0

General: Initial version . . . U1

2004/09/21 Version 1.1

General: configuration file names in lowercase (suggested by *Harald Harders*) . . . 81

remove 8-bit characters from the configuration files (suggested by *Harald Harders*) . . . 145

Protrusion: add factors for some more characters 153

settings for Adobe Minion (contributed by *Harald Harders*) . . . 154

`\DeclareCharacterInheritance`: new command: possibility to specify character inheritance . . . 115

`\MT@declare@sets`: remove spaces around set name 101

`\MT@find@file`: fix: also check whether the file for the base font family has already been loaded . . 81

`\MT@get@basefamily`: only remove suffixes ‘x’ or ‘j’ 82

`\MT@get@listname@`: don’t check for empty attributes list . . . 83

`\MT@ifempty`: fix: use category code 12 for the percent character (reported by *Tom Kink*) . . . 20

`\MT@is@number`: numbers may also be specified in hexadecimal or octal (suggested by *Harald Harders*) . . . 88

`\MT@pdftex@no`: fix: version check (reported by *Harald Harders*) . . . 15

`\MT@permute`: don’t use sets for empty encoding . . 117

`\MT@setup@expansion`: issue an error instead of a warning, when pdfTEX version is too old for `autoexpand` . . . 134

`\MT@split@codes`: fix: allow zero and negative values 45

`\MT@use@set`: remove spaces around set name . . . 105

2004/10/03 Version 1.2

Font aliases: declare `cmor` as an alias of `cmr` . . . 142

Font sets: new: `allmath` and `basicmath` . . . 141

Protrusion: add settings for Computer Modern Roman and Adobe Garamond in `TS1` encoding . . . 184

add settings for Computer Modern Roman math symbols . . . 189

`\MT@familyalias`: define alias font name as an alternative, not as a replacement . . . 41

`\MT@get@basefamily`: also remove ‘w’ (swash capitals) . . . 82

`\MT@get@highlevel`: check whether defaults have changed . . . 101

`\MT@get@inh@list`: fix: set inheritance list `\global`ly to `\@empty` . . . 84

`\MT@get@listname@`: alternatively check for alias font name . . . 83

`\MT@get@size`: additional magic to catch some errors 103

`\MT@get@size@@`: hijack `\set@fontsize` instead of `\@setfontsize` . . . 103

`\MT@loop`: fix: new macro, used instead of `\loop` . . 24

`\MT@maybe@do`: also check for alias font name . . . 41

`\MT@permute@@@@@@`: more sanity checks for `\SetProtrusion` and `\SetExpansion` . . . 118

`\MT@setupfont`: also search for alias font file . . . 38

fix: call `\@@enc@update` if necessary . . . 38

2004/10/27 Version 1.3

General: fix: specifying `load` option does no longer require to give a `name`, too . . . 112

Font aliases: declare `aer`, `zer` and `hfor` as aliases of `cmr` . . . 142

`\MT@fix@catcode`: check some category codes (compatibility with `german`) . . . 5

`\MT@load@list`: check whether list exists . . . 80

2004/11/12 Version 1.4

General: check for `pdfcprot` . . . 30

don’t use scratch registers in global definitions . . 85

use `\pickup@font` instead of `\define@newfont` as the hook for `\MT@setupfont` . . . 94

use one instead of five counters . . . 26

Protrusion: tweak quote characters for `cmr` variants

  - (OT1, T1, lmr) 159
- \microtypesetup: fix: set the correct levels, and remember them; warning when enabling an option disabled in package options 127
- \SetExpansion: fix: specifying extra options does no longer require to give a name, too 109

2004/11/17 Version 1.4a

- General: new option: final 123
- \MT@cfg@catcodes: fix: reset some more catcodes when reading files (reported by *Michael Hoppe*) 82

2004/11/26 Version 1.4b

- General: fix: set catcodes before reading global configuration file (reported by *Christoph Bier*) . . 126
  - optimisation: use less \expandafters and \csnames 19
- Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl) 153
  - slanted like italics 164
- \MT@checklist@family: fix: don't try alias family name if encoding failed 42
- \MT@get@basefamily: fix: failed for font names of the form abczz (reported by *Georg Verweyen*) 82
- \MT@get@slot: don't define \MT@char globally (save stack problem) 85
- \MT@ifdimen: don't set \MT@count globally (save stack problem) 21
- \MT@setup@PDF: new message if \pdfoutput is changed 131
- \MT@use@set: don't use undeclared font sets 105

2004/12/15 Version 1.5

- General: defaults: step: 4 (suggested by *Hàn Thế Thành*) 124
  - new option: selected, by default false (suggested by *Hàn Thế Thành*) 122
- Documentation: add 'Short history' U32
- Inheritance: remove \ss from T1 list, add \DJ . . . 146
- Protrusion: settings for Bitstream Charter 154
- \DeclareMicrotypeAlias: remove spaces around arguments 106
- \MT@cfg@catcodes: reset catcode of '=' (compatibility with Turkish babel) 82
- \MT@fix@catcode: reset catcode of '^' (compatibility with chemsym) 5
- \MT@get@highlevel: don't test defaults if called after begin document 101
- \MT@scale@factor: warning for factors outside limits 47
- \MT@scale@to@em: don't use \lpcode and \rpcode for the calculation 46
- \MT@set@ex@codes: allow non-selected font expansion 61
- \MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters 44
- \MT@setup@expansion: defaults: calculate step as min(stretch,shrink)/5 133
  - defaults: turn off expansion for DVI output 132
  - disable automatic expansion for DVI output . . . 133

2005/01/24 Version 1.6

- General: defaults: turn off expansion for old pdfTeX versions 126
  - load a font if none is selected 38
  - new option: factor, by default 1000 124
  - restructure dtx file 141
  - test whether \pickup@font has changed 97
  - test whether numeric options receive a number 124
  - use e-TeX's \ifcsname and \ifdefined if defined 19
- Protrusion: add italic uppercase Greek letters . . . 164
  - improve settings for numbers (pointed out by *Peter Muthesius*) 155
  - tune CMR math letters (OML encoding) 189
- \MT@get@charwd: use e-TeX's \fontcharwd, if available 46
- \MT@get@inh@list: correct message if selected is false 84
- \MT@set@ex@codes: introduce factor option 61
- \MT@set@pr@codes: introduce factor option 44
- \MT@setup@expansion: disable automatic expansion for old pdfTeX versions 134
- \MT@use@set: retain current set if new set is undeclared 105
- \MT@vinfo: new macro instead of \ifMT@verbose . . . 6

2005/02/02 Version 1.6a

- Documentation: add table of fonts with tailored protrusion settings U21
- \MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem reported by *Bernard Gaulle*) 85
- \MT@pdftex@no: new macro 14
- \MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions 62

2005/03/23

Version 1.7

- General: allow specification of size ranges (suggested by *Andreas Bühmann*) 102
  - disallow automatic expansion if pdfTeX too old 114
  - fix: remove space after `autoexpand` 114
  - new value for `verbose` option: `errors` 124
  - shorter command names 26
  - warning when running in draft mode 131
- Documentation: add hint about compatibility U28
  - remove table of match order (now table 1 on page 83) U11
- Protrusion: fix: remove \ from `OT1`, add `\textbackslash` to `T1` encoding 157
- `\LoadMicrotypeFile`: new command (suggested by *Andreas Bühmann*) 107
- `\Microtype@Hook`: new command for font package authors 127
- `\microtypesetup`: fix: warning also when setting to `(no)compatibility` 127
- `\MT@begin@catcodes`: also use inside configuration commands 82
- `\MT@cfg@catcodes`: reset catcode of '`:`' (compatibility with `french*` packages) 82
- `\MT@DeclareMicrotypeAlias`: may also be used inside configuration files 107
- `\MT@get@listname@`: use `\@tfor` (*Andreas Bühmann*'s idea) 83
- `\MT@get@slot`: remove backslash hack 85
  - test for `\chardefed` commands 86
  - test whether `\`⟨*encoding*⟩`\`⟨...⟩ is defined 85
- `\MT@if@list@exists`: don't define `\MT@pr@c@name` etc. `\globally`, here and elsewhere 84
- `\MT@ifdimen`: comparison with 1 to allow size smaller than 1 (suggested by *Andreas Bühmann*) 21
- `\MT@increment`: use e-TeX's `\numexpr` if available 26
- `\MT@is@composite`: new macro: construct command for composite character; no uncontrolled expansion 92
- `\MT@scale`: new macro: use e-TeX's `\numexpr` if available 26
- `\MT@set@ex@codes`: two versions of this macro 61
- `\MT@split@name`: don't define `\MT@encoding` &c. `\globally` 41
- `\MT@test@ast`: make it simpler 101
- `\MT@try@order`: always check for size, too (suggested by *Andreas Bühmann*) 83
  - fix: also check for `//`⟨*series*⟩`/`⟨*shape*⟩`//` (reported by *Andreas Bühmann*) 83
- `\MT@warn@code@too@large`: new macro: type out maximum protrusion factor 48
- `\MT@warn@err`: new macro: for `verbose=errors` 6
- `\showhyphens`: modify `\showhyphens` 135

2005/06/23

Version 1.8

- General: `\SetProtrusion`: new key: `unit` 113
  - if font substitution has occurred, set up the substitute font, not the selected one 94
  - new option: `config` to load a different main configuration file 126
  - new option: `unit`, by default `character` 125
- Documentation: add example for `factor` option U12
  - add example of how to get rid of a widow (suggested by *Adam Kucharczyk*) U14
  - add hint about error messages U28
- Font aliases: declare `pxr` and `txr` as aliases of `ppl` resp. `ptm` 143
- Font sets: add `U` encoding to `allmath` 141
- Inheritance: remove `\DJ` from `T1` list (it's the same as `\DH`) 146
- Protrusion: add LY1 characters for Times 163
  - settings for AMS math fonts 192
  - verified settings for slanted Computer Modern Roman 174
- `\add@accent`: fix: disable micro-typographic setup inside `\add@accent` (reported by *Stephan Hennig*) 97
- `\DeclareMicrotypeAlias`: warning when overriding an alias font 106
- `\DeclareMicrotypeSetDefault`: new command: set default font set 105
- `\MT@cfg@catcodes`: reset catcodes of the remaining ASCII characters 82
- `\MT@check@rlist`: made recursive 119
- `\MT@curr@list@name`: new macro: current list type and name 93
- `\MT@declare@sets`: warning when redefining a set 101
- `\MT@define@set@key@`: use comma lists instead of token lists 101
- `\MT@find@file`: no longer wrap names in commands 81
- `\MT@fix@fontdimen@six`: new macro: test whether `\fontdimen` 6 is defined 40
- `\MT@get@charwd`: warning for missing (resp. zero-width) characters 46
- `\MT@get@listname@`: made recursive 83
- `\MT@get@slot`: fix: expand active characters 85
  - test whether `\`⟨*encoding*⟩`\`⟨...⟩ is defined made more robust 85
- `\MT@get@unit`: new macro: get unit for codes 49
- `\MT@in@rlist`: made recursive 24
- `\MT@is@active`: new macro: translate `inputenc`-defined characters 89
- `\MT@is@letter`: warning for non-ASCII characters 88
- `\MT@ledmac@setup`: character protrusion with `ledmac` 28
- `\MT@map@clist@n`: new macro: used instead of `\@for` 23
- `\MT@map@tlist@n`: new macro: used instead of `\@tfor` 23
- `\MT@old@cmd`: renamed commands from `\..MicroType..` to `\..Microtype..` 6
- `\MT@pdftex@no`: case 5: pdfTeX 1.30 14
- `\MT@permute@@@@@@`: add ranges to the beginning of the lists 119
- `\MT@scale`: fix: remove spaces in ɇ-TeX variant (reported by *Mark Rossi*) 26
- `\MT@setupfont@hook`: restore `\%` and `\#` when `hyperref` is loaded 30
  - restore `csquotes`'s active characters 30
  - restore percent character if Spanish `babel` is loaded 30
- `\MT@split@codes`: get character width once only 45
- `\MT@use@set`: fix: remove braces in first line 105
- `\MT@xadd`: simplified 23

2005/10/28 Version 1.9

General: \DeclareMicrotypeSet: new key: font . 104
\SetProtrusion: value 'relative' renamed to 'character' for key unit 113
allow context-specific font setup 94
compatibility with TeX Live hack (reported by *Herbert Voß*) 13
disable microtype setup inside hyperref's \pdfstringdef (reported by *Hàn Thế Thành*) . . 31
fix: use true as the default value 121
option unit: rename value relative to character 125
Documentation: add remark about Type 1 fonts required for automatic font expansion U7
Font aliases: declare qpl and qtm (qfonts, TeX Gyre) as aliases of ppl resp. ptm 143
Font sets: add OT4 encoding to text sets 141
add T5 encoding to text sets 141
Inheritance: add list for OT4 147
add list for T5 (requested by *Hàn Thế Thành*) . . 148
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR 157
settings for OT4 encoding (Computer Modern Roman, Palatino, Times) 153
settings for T5 encoded Computer Modern Roman 153
\DisableLigatures: new command: disable ligatures (requires pdfTeX 1.30) 107
\microtypecontext: new command: change setup context in the document 98
\MT@checklist@family: fix: add two missing \expandafters 42
\MT@detokenize@c: fix the ε-TeX version 20
\MT@exp@two@n: new macros: less \expandafters . . 19
\MT@get@opt: new key 'preset' to set all characters to the specified value before loading the lists 49
\MT@is@active: redone: use \set@display@protect 89
\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning 88
\MT@maybe@do: redone 41
\MT@rem@from@clist: new macro: remove an item from a comma list 24
\MT@scale@factor: generalised 47
\MT@setup@expansion: disable expansion if both step and shrink are zero 134
warning if user requested zero step 133
\MT@toks: use instead of \toks@ 16
\SetProtrusion: (et al.) new key: font 108

2005/12/05 Version 1.9a

General: '⟨*file name*⟩/⟨*line number*⟩' as default list name 112
new option: defersetup, by default true 123
remove superfluous test whether \pickup@font has changed 97
Documentation: add explanation for error message in DVI mode U29
add explanation for error message with non-Type 1 fonts U30
Font aliases: declare mdbch (mathdesign) as an alias of bch 144
Protrusion: fix: remove '_' from OT1 encoding . . . 158
settings for T5 encoded Charter 153
\microtypesetup: inside the preamble, accepts all package options 127
\MT@check@font@cx: optimise context-sensitive setup 97
\MT@define@set@key@: don't expand variables immediately (requested by *Georg Verweyen*) 101
\MT@get@highlevel: no longer check whether defaults have changed 101
\MT@ifdefined@c@T: new macros: true case only . . 20
\MT@ifint: use \pdfmatch if available 20
\MT@ifstreq: use \pdfstrcmp if available 22
\MT@in@clist: fix 23
\MT@info@missing@char: info instead of warning (after *Michael Hoppe* reported that the 'fl' ligature is missing in Palatino SC) 47
\MT@is@feature: new macro: check for pdfTeX feature 27
\MT@map@clist@n: following LaTeX3 23
\MT@permute@@@@@: don't define permutations for unused encodings 118
\MT@rem@from@clist: fix 24
\MT@setup@: defer setup until the end of the preamble 27

2006/01/20 Version 1.9b

General: compatibility with listings: sanitise more catcodes (reported by *Holger Uhr*) 32
compatibility with the extendedchar option of the listings package 32
Documentation: activate expansion in the distributed PDF U1
add samples of micro-typographic features U3
\MT@features: use throughout the package to adjust to beta-ness 27
\MT@ifdimen: use \pdfmatch if available 21
\MT@warn@code@too@large: fix calculation with present factor 48

2006/02/02 Version 1.9c

Documentation: add example of how to increase protrusion of footnote markers (suggested by *Georg Verweyen*) U22
Protrusion: settings for URW Garamond 154
\MT@define@code@key@font: fix: context was ignored 111
\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in v1.9b) 111

2006/05/05 Version 1.9d

- Font sets: md* instead of m series in basic sets 141
  - add QX encoding to text sets 141
- Inheritance: add list for QX encoding (contributed by *Maciej Eder*) 148
- Protrusion: settings for QX encoding (contributed by *Maciej Eder*) 161
  - settings for Euro symbols (Adobe, ITC, marvosym) 200
  - tweak AMS settings 192
- \DeclareCharacterInheritance: fix: empty context 115
- \MT@detokenize@n: new macro: use \detokenize if available 20
- \MT@get@ex@opt: fix: evaluate preset 63
- \MT@get@font@dimen: warning for zero fontdimen 47
- \MT@get@opt: optimise: don't reset when preset option is set 49
  - set list name before presetting 49
- \MT@is@active: support for Unicode (inputenc/utf8) 89
- \MT@setupfont@hook: restore \% and \# when tex4ht is loaded (reported by *Peter Dyballa*) 30
- \SetProtrusion: (et al.) optimise: unify keys for mandatory argument 108
  - (et al.) split keys of optional and mandatory argument 108

2006/07/28 Version 1.9e

- General: fix: default value for activate: true 122
- Documentation: add hint about unknown encodings U27
  - include LPPL 273
- Font aliases: declare zeur and zeus (eulervm) as aliases of eur resp. eus (euler) 144
- Inheritance: adapt to marvosym's changed encoding 150
- Protrusion: complete settings for Euler Fraktur and Script fonts 199
  - fix: forgotten comma in mt-mvs.cfg; adapt to marvosym's changed encoding 200
  - settings for Euler Roman font 196
- \DeclareCharacterInheritance: new key 'inputenc' to set the input encoding 115
- \MT@rem@from@clist: model after \@removeelement 24
- \MT@setup@: empty \MT@setup@ after use (compatibility with the combine class) 27
- \pickup@font: no tracing with trace package 96
- \SetExpansion: new key: inputenc 109
- \SetProtrusion: (et al.) new key: inputenc 108

2006/09/09 Version 1.9f

- Protrusion: fix: euler-vm did not load euler settings 197
- \MT@curr@list@name: fix: \MessageBreak must not be expanded 93
- \MT@gdef@n: new macros: global variants 19
- \MT@get@inh@list: fix: input encoding must be set after the inheritance list has been parsed 85
- \MT@glet: new macro 19
- \MT@reset@context: only reset context if it has actually been changed 99
- \MT@set@inh@list: fix: forgotten comma in the features list 115
- \MT@set@named@keys: new macro: set name first, simplify parsing of optional argument 110
- \SetProtrusion: (et al.) set catcodes before parsing optional argument 108

2007/01/14 Version 2.0

- General: compatibility with listings: set catcode of backslash to zero (reported by *Steven Bath*) 32
  - compatibility with soul: register \textls and \lsstyle 32
  - new option: babel, by default false (language-dependent setup suggested by *Ulrich Dirr*) 122
  - new option: letterspace, by default 100 124
  - new package letterspace: a stripped-down version, containing the letterspacing commands only U1
  - option 'babel': fix: switch off French babel's shorthands properly (reported by *Daniel Flipo*) 139
  - option 'babel': switch off Turkish babel's shorthands 139
  - option 'unit', \SetProtrusion: deprecate value 'relative' completely 113
- Documentation: add hint about how to increase font_max and font_mem_size U30
  - add hint about warning when tracking *and* expansion is applied to a font U30
  - add remark about 'disable' (previously draft) option disabling microtype (noted by *Michalis Miatidis*) U8
  - qualify hint about web documents with regard to older pdfTeX versions U27
  - qualify hints about expansion error messages with regard to older pdfTeX versions U29
- Font sets: new: footnotesize and scriptsize 141
  - new: smallcaps 141
- \DeclareMicrotypeBabelHook: new command: interaction with babel 108
- \lsstyle: fix: font switches don't pose a problem anymore 72
  - fix: letterspacing commands may be nested 72
  - new command: letterspacing 72
  - totally redone, using the new \letterspacefont 72
- \MT@declare@sets: fix: empty size list when redefining set 101
- \MT@is@symbol: made even more robust 90
- \MT@load@inputenc: sanitise catcodes before loading input encoding (problem with listings) 50
- \MT@pdftex@no: case 6: pdfTeX 1.40 14
- \MT@setup@noligatures: maybe disable \MT@noligatures after the preamble 138

- `\MT@split@name`: adjust to possible letterspacing 41
- `\SetExtraKerning`: new command: additional kerning 110
- `\SetExtraSpacing`: new command: adjustment of interword spacing 109
- `\SetTracking`: new command: tracking 109
- `\textls`: new command: letterspacing 77
  - starred version: remove spaces around text 77
- `\tracingmicrotypeinpdf`: new debug method: mark all fonts with PDF annotations 7

2007/01/21

Version 2.1

- General: compatibility with `pinyin`: disable `microtype` in `\py@macron` (reported by *Sven Naumann*) 32
  - fix: `letterspace` package forgot to load `keyval` 16
- `\lslig`: new command: protect ligatures in letterspaced text 72
- `\MT@get@ls@basefont`: redone: use `\pdfmatch` to make it bullet-proof 73
- `\MT@orig@pickupfont`: compatibility with `CJK`: also check for its definition 95
- `\textls`: fix: use `\hmode@bgroup` 77

2007/07/14

Version 2.2

- General: disable `microtype` if `wordcount` is loaded (reported by *Ross Hetherington*) 27
  - new option: `copyfonts` 123
  - simplify key declarations 112
  - use catcode trickery for e-TeX test 13
- Documentation: add hint about error message with pdfTeX 1.40 U29
  - add hint about extra TOC leader dot (first discovered by *Morten Høgholm*) U27
  - add overview U4
  - logo transparency and amusement U1
- Font aliases: declare `chr` (`chmath`) as an alias of `bch` (reported by *Geoff Vallis*) 144
  - declare `fp9x`, `fp9j` (FPL Neu) as aliases of `ppl[xj]` 143
- Font sets: default set for tracking: `smallcaps` 142
- Inheritance: remove '-' → '127' 146
- Protrusion: settings for Bitstream Letter Gothic 154
- Spacing: add sample 201
- Tracking: add ligatures that are to be disabled 151
- `\DeclareMicrotypeVariants`: new command 106
- `\DisableLigatures`: new optional argument: disable selected ligatures only 107
- `\lslig`: always defined 72
- `\MT@checklist@font`: fix: construct font name from characteristics 43
- `\MT@copy@font`: optionally work on copies of fonts 39
- `\MT@get@basefamily`: redone, working on font names and suffixes of arbitrary length 82
- `\MT@get@charwd`: subtract letterspacing amount from width 47
- `\MT@get@ls@basefont`: fix again: remember base font in a macro 73
- `\MT@ifdimen`: employ LuaTeX features if available 21
- `\MT@ifint`: employ LuaTeX features if available 20
- `\MT@ifstreq`: employ LuaTeX features if available 22
  - fix: e̸-TeX version shouldn't use `\x` and `\y` (found by *Wiebke Petersen*) 22
- `\MT@is@composite`: more robust: expand exactly once 92
- `\MT@is@symbol`: expand once more (for `frenchpro`) 91
- `\MT@lsfont`: use `\font@name`, not `\MT@font` 67
- `\MT@lua`: (basic) support for LuaTeX 16
- `\MT@pdftex@no`: case 7: pdfTeX 1.40.4 14
- `\MT@preset@aux@space`: generalised 51
- `\MT@set@all@pr`: (et al.) allow empty values 44
- `\MT@set@inputenc@`: only load `inputenc` files if necessary 49
- `\MT@set@tr@codes`: disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4) 68
  - possibility to customise interword spacing 68
- `\MT@setup@expansion`: warning if `stretch` or `shrink` aren't multiples of `step` 134
- `\MT@setupfont`: don't call `\@@enc@update` anymore 38
  - only add features that are available with the respective pdfTeX 38
- `\MT@setupfont@hook`: restore percent character if Galician `babel` is loaded 30
- `\MT@the@pr@code@tr`: adjust protrusion of letterspaced fonts 45
- `\MT@tracking`: remember fonts that shouldn't be letterspaced 66
- `\MT@tracking@`: fix: tracking couldn't be re-enabled 66
- `\MT@warn@tracking@DVI`: warning when letterspacing in DVI mode 137
- `\MT@with@babel@and@T`: also inspect class options 28
- `\pickup@font`: `letterspace`: setup inside group 96
- `\SetTracking`: new key '`no ligatures`' to disable ligatures of letterspaced fonts 109
  - new keys '`spacing`' and '`outer spacing`' to adjust interword spacing (suggested by *Steven E. Harris*) 109
  - third argument may be empty 109
- `\textmicrotypecontext`: new command: wrapper around `\microtypecontext` 99

2007/12/23

Version 2.3

- General: disable `\microtypecontext` in `hyperref`'s `\pdfstringdef` 31
  - fix: really switch off Turkish shorthands 139
  - new value for `verbose` option: `silent` (suggested by *Karl Berry*) 124
  - turned some warnings into errors 124
- Documentation: add kerning sample U18
  - add letterspacing illustration U16
- `\do@subst@correction`: remember substitute font for all times (reported by *Stephan Hennig*) 96
- `\lslig`: redone: extract outer kerns from current letterspacing amount 72

\microtypecontext: made robust (reported by *Stephan Hennig*) 98

\MT@begin@catcodes: fix: don't disable \KV@@sp@def 82

\MT@define@set@key@font: font: single asterisk means normal font 104

\MT@in@clist: don't use \x (reported by *Peter Meier*) 23

\MT@is@active: support for extended Unicode (inputenc/utf8x resp. ucs) – experimental 89

\MT@noligatures: fix: set evaluation didn't work (bug introduced in v2.2) 79

\MT@plain: letterspace: support for eplain/miniltx 12

\MT@set@curr@os: adjusting spaces made more reliable 68

\MT@set@tr@codes: also adjust tracking if protrusion is not enabled, and even for letterspace (reported by *Stephan Hennig*) 68

possibility to customise outer kerning (suggested by *Stephan Hennig*) 68

\MT@SetTracking: sanity check for value 109

\MT@setup@tracking: enable protrusion when tracking is enabled 135

\MT@tr@outer@l: only change pre outer space if it contains shrink 75

2008/02/29 Version 2.3a

General: fix test for soul under plain TeX 32

Documentation: add hint about babel having to be loaded first U28

add table of available and enabled features U6

Protrusion: adjust LMR quotation marks again 160

\MT@error@doesnt@work: error messages if pdfTeX is too old for extensions 136

\MT@fix@catcode: fix catcodes earlier, and also for the letterspace package 5

\MT@getkey: fix: key=val in class options list 131

\MT@set@codes: generalised 45

\MT@setupfont@hook: restore percent character if Mexican babel is loaded 30

2008/06/04 Version 2.3b

\MT@exp@gcs: new macro: reduce save stack size 19

\MT@font@copy: enable font copies also with protrusion contexts (reported by *Nathan Rosenblum*) 39

\MT@get@size@@: grouping 103

\MT@noligatures@: fix: warning messages for unknown slots 79

\MT@orig@pickupfont: compatibility with CJKutf8: also check for its definition 95

\MT@requires@latex: new macro 13

\MT@set@tr@codes: fix: protrusion adjustment only for new fonts (reported by *Wolfram Schaalo*) 68

\MT@tr@outer@l: fix: only in horizontal mode 75

make \spaceskip-aware (ragged2e) 75

\MT@tr@outer@r@: additional test for horizontal mode 76

2008/11/11 Version 2.3c

General: LuaTeX supported by default 15

Documentation: add hint about spacing being experimental U28

add hint about partial incompatibility with CJK U28

Inheritance: add \textcommabelow[STst] to QX encoding (reported by *Vasile Gaburici*) 148

\MT@detokenize@c: fix: remove last space only (reported by *Ulrich Dirr*) 20

\MT@tr@outer@r@: additional test for horizontal mode (reported by *Sveinung Heggen*) 76

2009/03/27 Version 2.3d

General: fix pinyin compatibility check (reported by *Silas S. Brown*) 32

move setup to the very end (for *Colin Rourke*) 139

\ifMT@inannot: use pdftexcmds for debugging 7

\lsstyle: disable for LuaTeX 72

make invalid in math mode 72

\microtypesetup: select font after setup 127

\MT@check@active@set: warning for missing default sets 127

\MT@lua: update for LuaTeX 0.36 16

\MT@set@tr@codes: allow zero tracking 67

\MT@set@tr@zero: fix: allow switching off tracking (reported by *Ulrich Dirr*) 73

\MT@setup@expansion: default step: 1 for pdfTeX versions ≥ 1.40 133

\MT@tr@outer@r@: don't use \x (reported by *Ulrich Dirr*) 76

fix: don't adjust in math mode (reported by *Christoph Bier*) 76

fix: don't adjust inside discretionary (reported by *Maverick Woo*) 76

\MT@tr@set@okern: allow empty value for outer kerning 78

\textls: make math mode aware 77

2009/11/09 Version 2.3e

Expansion: settings for T2A encoding (contributed by *Karl Karlsson*) 152

Font sets: sc* instead of sc in smallcaps set 141

add T2A encoding 141

Protrusion: settings for T2A encoding (contributed by *Karl Karlsson*) 160

Spacing: settings for T2A encoding (contributed by *Karl Karlsson*) 202

\MT@fix@fontdimen@six: fix: gobbling settings with tracking failed (reported by *Leo*) 40

\MT@setup@: make space-unaware (requested by *Marcin Borkowski*) 27

\MT@tikz@setup: compatibility with tikz (first reported by *Christian Stark*) 30

\MT@tr@outer@r@: fix: set current kerning and spacing again (found by *Lars Rönnbäck*) 76

2010/01/10 Version 2.4

General: new file microtype.lua containing the lua functions (contributed by *Élie Roux*) 18

Protrusion: settings for T2A encoded Minion (contributed by *Karl Karlsson*) 160

2013/03/13 Version 2.5

General: allow contexts for LuaTeX 112
- disable 'DVIoutput' option for XeTeX 122
- fix: check whether '⟨*file*⟩/⟨*line*⟩' list name already exists (reported by *Till A. Heilmann*) 112
- letterspacing with LuaTeX 0.62 66
- new files: microtype-pdftex.def, microtype-xetex.def, microtype-luatex.def, containing engine-specific definitions 13
- protrusion with XeTeX 15
- restore \space inside listings (reported by *Rolf Dieterich*) 32

Documentation: add hint about LuaTeX compatibility U28
- add hint about spacing and ragged2e U28
- add hint about dtx source code U30
- include microtype-logo.dtx and microtype-lssample.dtx 238

Font aliases: declare lmsy and lmm as aliases of cmsy resp. cmm (reported by *Jonas Hogstrom*) 142
- declare zgmx etc. (garamondx) as aliases of ugm 144
- declare Latin Modern Roman (OpenType version) as alias of lmr when fontspec is loaded 142
- declare TeX Gyre Pagella, Asana Math, Palatino LT Std, and Palatino as aliases of Palatino Linotype (OpenType version) 143

Font sets: add EU1 and EU2 encodings 141

Inheritance: add rudimentary list for EU1 and EU2 149

Protrusion: add default lists for EU1 and EU2 158
- improvements to Computer Modern Roman italics (contributed by *Hendrik Vogt*) 164

Tracking: add EU2 encoding to default list 151

\DeclareCharacterInheritance: allow more than one encoding 115

\DeclareMicrotypeAlias: ignore spaces 106

\ifMT@nofamily: info if settings are not family-specific (suggested by *Hàn Thế Thành*) 43

\LoadMicrotypeFile: remove all spaces in font name 107

\lsstyle: fix: ensure to set up math fonts (reported by *RazorXsr*) 72

\MT@define@code@key@family: compatibility with fontspec: remove its internal counter (reported by *Till A. Heilmann*) 110

\MT@define@code@key@font: scrub fontspec feature count (found by *Meho R*) 111

\MT@do@font: adapt for LuaTeX 25
- adapt for XeTeX 26

\MT@get@slot@: adapt for LuaTeX (requested by *Georg Duffner*) 86
- adapt for XeTeX 86

\MT@if@outer@next: fix: conflict with amsmath (reported by *Scott Pakin*) 75

\MT@info@missing@char: fix error message for XeTeX (reported by *Juan Acevedo*) 47

\MT@is@charx: compatibility with xunicode 91

\MT@ledmac@setup: fix to work with XeTeX (reported by *Maïeul Rouquette*) 28

\MT@ls@set@ls: allow formulas in optional argument to \textls (fix by *Heiko Oberdiek*) 77

\MT@microtypecontext: fix: ensure to set up math fonts (reported by *RazorXsr*) 99

\MT@register@subst@font: only register substituted font if it isn't registered already (reported by *George Gratzer* and *Josep Maria Font*) 97

\MT@register@subst@font@cx: only register if it isn't registered already 98

\MT@scrubfeatures: compatibility with fontspec: remove its internal counter 41

\MT@set@all@pr: fix: remove space (found by *Meho R*) 44

\MT@set@pr@codes: make info about generic settings encoding-specific (reported by *Sebastian Schubert*) 44

\MT@setup@spacing: warning with ragged2e (reported by *Steffen Hoffmann*) 136

\MT@setupfont: select font with fontspec (found by *Georg Duffner*) 38

\MT@setupfont@hook: restore \% and \# when mathastext is loaded (found by *Seamus Bradley*) 30

2013/05/23 Version 2.5a

General: use luatexbase instead of luatextra (contributed by *Élie Roux*) 18

Documentation: add notes on typesetting the documentation U30
- include OpenType configuration files 206

\MT@afteraftergroup: fix: get outer kerning and spacing of nested letterspacing right 69

\MT@get@slot@: adapt to luaotfload v2.2 (contributed by *Élie Roux*) 87

\MT@led@unhbox@line: simplified 28

\MT@ledmac@setup: support for eledmac 28

\MT@ls@outer@k: add marker for tightly nested letterspacing 78

\MT@set@tr@codes: fix: load font for fontspec 68

\MT@xspace: fix outer spacing problem with xspace (reported by *Dave*) 77

2016/05/01

Version 2.6

- General: load `luaotfload` with LuaTeX 18
  - redefine `\MT@setupfont@hook` globally for problem with `tikzposter` (reported by *Sam Mason*) 31
- Documentation: add hint about partial incompatibility with `xeCJK` and `luatexja` U28
  - missing characters printed with Charis SIL 206
- Font sets: add `TU` encoding (notified by *Will Robertson*) 141
  - add `si` and `scit` to `smallcaps` set (reported by *uli*) 141
  - new: `allmath-nott` and `alltext-nott` (suggested by *Karl Berry*) 141
- Inheritance: add `TU` encoding 149
- Protrusion: add `TU` encoding to lists 158
- Tracking: add `TU` encoding to default list 151
- `\DeclareMicrotypeSet`: ignore spaces 100
- `\DeclareMicrotypeSetDefault`: ignore spaces 105
- `\DeclareMicrotypeVariants`: ignore spaces 106
- `\lsstyle`: fix: ensure to set up math fonts (reported by *kleenstar*) 72
- `\microtypecontext`: allow `activate` shortcut (reported by *Karl Berry*) 98
- `\MT@declare@sets`: fix: undefine lists for redefining 101
- `\MT@do@font`: speed up for LuaTeX 25
- `\MT@engine`: fix test with LuaTeX 0.85 13
- `\MT@get@slot@`: fix: could fail with XeTeX (reported by *Christopher Schramm*) 86
- `\MT@is@xchar`: update for `fontspec`'s `TU` encoding 91
- `\MT@ledmac@setup`: support for `reledmac` 28
- `\MT@luatex@no`: update for LuaTeX 0.85 (renamed primitives) 15
- `\MT@noligatures@`: use `luaotfload` function to keep/inhibit ligatures 79
- `\MT@orig@pickupfont`: (in)compatibility with `luatexja`: disable unknown slots warnings (reported by *Max*) 95
  - (in)compatibility with `xeCJK`: disable unknown slots warnings (reported by *HcN*) 95
  - compatibility with `xeCJK`: pretend that `CJK` wasn't loaded 95
- `\MT@set@tr@codes`: use `luaotfload`'s `kernfactor` feature if available 67
- `\MT@xspace`: fix `outer spacing` problem with (not only) `algorithm` (reported by *Henning* and *Ronnie Marksch*) 77
- `\UseMicrotypeSet`: ignore spaces 105

2016/05/14

Version 2.6a

- General: fixes for `letterspace` package with LuaTeX 25
- `\MT@do@font`: fix `lua` function (reported by *Herbert Voß*) 25
- `\MT@ls@fontspec@font`: fix for value of ±1000 70

2017/07/07

Version 2.7

- General: drop `luatexbase` with recent LaTeX 18
  - warning with `minimal` class 27
- Documentation: mention that additional kerning does not work in math mode (discovered by *'Daniel'*) U17
- Font aliases: declare aliases for `newpx` 143
  - declare aliases for `newtx` 143
  - declare aliases for `tempora` 143
  - declare aliases for `XCharter` 144
  - declare Latin Modern Roman as alias of `lmr` with new LaTeX format (reported by *Ulrike Fischer*) 142
- Protrusion: automatically choose correct names for Charis SIL small caps (reported by *'ltcomdata'*) 227
- `\lsstyle`: fix: prevent infinite loop with `psnfss` and `exscale` packages (reported by *user11126*, solution by *Ulrike Fischer*) 72
- `\MT@check@range@`: don't warn for override if conflicting list is loaded 120
- `\MT@is@composite`: compatibility with LaTeX 2017/01/01 (`\DeclareUnicodeComposite`) (reported by *Ulrike Fischer* and *'jcr'*) 92
- `\MT@ls@fontspec@font`: fix for '`file:`⟨*font*⟩' spec (reported by *Reinhard Kotucha*) 70
- `\MT@permute@@@@@@`: don't warn for override if conflicting list is loaded 119
- `\MT@reset@ef@codes`: only reset `\efcodes` for older LuaTeX versions 62
- `\MT@setup@expansion`: don't disable automatic expansion for DVI output with LuaTeX 133
- `\MT@tikz@setup`: compatibility with `tikz` (again) 30
- `\MT@warn@tracking@DVI`: don't warn for letterspacing in DVI mode with LuaTeX 137

2018/01/14

Version 2.7a

- General: disallow non-automatic expansion with LuaTeX 114
- `\MT@auto`: remove '`autoexpand`' for LuaTeX 1.0.6 (reported by *Ulrike Fischer*) 133
  - with LuaTeX, non-automatic font expansion is no longer possible (as confirmed by *Hans Hagen*) 133
- `\MT@get@highlevel`: test whether `\...default` is defined 101
- `\MT@get@slot`: expand active characters earlier 85
- `\MT@info@notracking@`: defer 'No tracking' message 42
- `\MT@is@active`: compatibility with `newunicodechar` (reported by *Nils Anders Danielsson*) 89

2019/02/28

Version 2.7b

- General: update lua function `microtype.info` after changes in `luaotfload` (reported by *Moritz Wemheuer* and *Ulrike Fischer*) 18
- Documentation: update hint about non-7-bit characters (notified by *Frank Mittelbach*) U28
- Inheritance: add textquotedblleft ligature to OT4 (reported by *Franz Wexler*) 147
- `\MT@info@missing@char`: fix message for glyphs specified as names in XeTeX (reported by *Paolo Ney*) 47
- `\MT@setupfont`: always select current font with XeTeX and LuaTeX (reported by *Paolo Ney*, solution by *Ulrike Fischer*) 38

2019/10/10

Version 2.7c

- General: turn warning into info when overwriting the `keepligature` function (reported by *Andy N*) 80
- `\MT@is@active`: compatibility with LaTeX 2019/10/01 89
- `\MT@is@symbol`: take care of `\remove@tlig` 91
- `\showhyphens`: compatibility with LaTeX 2019/10/01 (reported by *Phelype Oleinik* and *Falk Hanisch*) 135

2019/11/18

Version 2.7d

- `\MT@copy@font@`: in LuaTeX, don't use the `\copyfont` primitive, but load the font anew (reported by *Paolo Polesana* and *Oliver Kopp*) 39
- `\MT@register@subst@font`: remove substitute font from list (reported by *Markus Kohm*) 97
- `\MT@register@subst@font@cx`: remove substitute font from lists 98

2020/12/07

Version 2.8

- General: `letterspace` works with e-TeX only 13
  - compatibility with `soul`: patch for font change (reported by *Md Ayquassar*) 32
  - fix for `luatexbase` 18
- Documentation: declare `DVIoutput` option deprecated U8
  - squash fake news about automatic font expansion with `dvilualatex` U7
- Font aliases: declare aliases for `step` and `domitian` (notified by *Daniel Benjamin Miller*) 143
  - declare aliases for `stix` and `stix2` fonts 143
  - declare New Computer Modern as an alias of Latin Modern Roman 143
- Font sets: default set for expansion: `alltext-nott` (suggested by *Aman Mehra*) 142
  - default set for spacing: `alltext-nott` 142
- `\lsstyle`: fix: enforce math setup, again 72
- `\microtypecontext`: fix `activate` shortcut 98
  - ignore spaces 98
- `\MT@do@font`: fix for XeTeX 26
  - simplify `lua` function 25
- `\MT@fix@fontdimen@six`: try to fix zero `\fontdimen` 6 40
- `\MT@if@opentype@font`: use `lua` function 25
- `\MT@ifstreq`: use XeTeX's `\stringcmp` 22
- `\MT@setup@expansion`: warning when expanding in DVI mode with LuaTeX (reported by *Daniel Benjamin Miller*) 132
- `\MT@tr@set@space@`: simplified 74
- `\MT@tr@unit@`: fix: allow unit regardless whether letterspacing is set 70
- `\textmicrotypecontext`: ignore spaces 99

2021/02/22

Version 2.8a

- General: rename `mt-pad.cfg` to `mt-EBGaramond.cfg` (requested by *Karl Berry*) 141
  - rename `mt-PalatinoLinotype.cfg` to `mt-Palatino.cfg` (requested by *Karl Berry*) 206
- Documentation: reorganise table 1 for clarification (suggested by *Daniel Benjamin Miller*) U6
- Font aliases: add `-LF`, `-TLF`, `-OsF` and `-TOsF` as variants 142
  - declare `mlmodern` fonts as aliases of Latin Modern (reported by *Daniel Benjamin Miller*) 142
  - reference New Computer Modern also by file name (reported by *Canageek*) 143
- Inheritance: specify 'ff' ligature as Unicode instead of glyph name 206
- Protrusion: hide `euroitc` settings (requested by *Karl Berry*) 200
- Spacing: add dummy settings (reported by *DORpapst*) 204
- `\DeclareCharacterInheritance`: skip settings for tracking 115
- `\MT@get@size@@`: compatibility with `svjour3` (reported by *Ekkehart Schlicht* (no relation!)) 103
- `\MT@glet`: use LuaTeX's `\glet`, if available 19

2021/02/25

Version 2.8b

- Inheritance: dummy settings for the Font Awesome font (`mt-FontAwesome.cfg`) 214
  - settings for the Lato font (`mt-Lato.cfg`) (reported by *dsedivec*) 214

2021/03/14 Version 2.8c

- `\ifMT@fontspec`: add to hook for `fontspec` 29
- `\MT@is@opt@char`: fix for optionally defined glyphs (reported by *Frank Mittelbach*) 91
- `\MT@remove@tlig`: fix for text commands containing conditionals (reported by *Frank Mittelbach*) 92

2021/10/31 Version 3.0

- General: `letterspace` loads `microtype.lua` 18
  - compatibility fix for `unicode-math` with `luainputenc` (reported by *tnull*) 33
  - move development to GitHub U1
  - new option: `disable`, replacing the `draft` option (suggested by *Frank Mittelbach*) 124
  - new options: `patch` and `nopatch` 125
  - new package: `microtype-show` for visual debugging 8
  - protrusion patch for TOC 35
  - protrusion patch for `\item` 34
  - protrusion patch for equation numbers (provided by *Holger Gerhardt*) 35
  - protrusion patch for footnote text 36
  - remove option `final` 123
  - require package `etoolbox` (for `patch` option) 16
- Documentation: split up in User manual (`microtype.pdf`) and Implementation doc (`microtype-code.pdf`) U1
  - add hint about unknown slot numbers U29
  - remove hint about extra TOC leader dot (fixed in LaTeX) U27
  - remove hint about web documents with pre-1.40 pdfTeX (it's been 14 years . . .) U27
  - rename option `draft` to `disable` (insistently requested by *Frank Mittelbach*) U8
  - split up table of fonts with tailored protrusion settings U22
- Font aliases: declare `basic` alias for Bergamo Std 144
  - declare `basic` aliases for the Fontin font 144
  - declare `empty` aliases for `fontawesome5` 144
  - New Computer Modern has its own settings 143
- Inheritance: add settings for LGR 150
  - generalise basic inheritance settings 214
  - generalise empty settings 214
  - settings for EB Garamond (OpenType) 210
  - settings for New Computer Modern (provided by *Antonis Tsolomitis*) 206
- Protrusion: LGR settings for EB Garamond 163
  - fix quotation marks in LMR 215
  - settings for EB Garamond (OpenType) 228
  - settings for EB Garamond figure variants 158
  - settings for EB Garamond small caps italics 182
  - settings for New Computer Modern (provided by *Antonis Tsolomitis*) 215
- `\ifMT@fontspec`: fix for changed hook name order (reported by *Frank Mittelbach* and *Karl Berry*) 29
- `\leftprotrusion`: new command 52
- `\MT@do@font`: guard against return values outside Unicode range (for HarfBuzz) 25
- `\MT@get@slot@`: convert floats to integers (reported by *azur*) 87
  - guard against return values outside Unicode range (for HarfBuzz) 87
- `\MT@has@inh@prefix`: new: automatical protrusion based on char widths difference (suggested by *Daniel Benjamin Miller*) 117
- `\MT@is@opt@char`: fix for incompatibility with `syntax` (reported by *Safron*) 91
- `\MT@ls@fontspec@font`: use `lua` to parse font spec/ fix for font spec containing spaces (reported by *Aman Mehra*) 70
- `\MT@old@cmd`: old command names will raise an error 6
- `\MT@pdftex@no`: case 8: pdfTeX 1.40.23 14
- `\MT@set@inh@list`: fix: grouping with `\begingroup` ... `\endgroup` instead of {. . . } (reported by *chsk*) 115
- `\MT@set@tr@codes`: with LuaTeX, tracking also works for fonts with a zero `\fontdimen` 6 67
  - with pdfTeX 1.40.23, tracking also works for fonts with a zero `\fontdimen` 6 67
- `\noprotrusion`: provide command for older LaTeX versions 52
- `\rightprotrusion`: new command 52

2021/12/02 Version 3.0a

- General: abort earlier if no capable engine found 16
  - allow `\microtypesetup{(no)patch=all}` 130
  - define function `math.tointeger` for older LuaTeX versions 18
  - fix protrusion patch `footnote` with KOMA classes (reported by *MisterFiLou*) [issue #2] 37
  - fix protrusion patch `footnote` with `hyperref` (reported by *Liang-Bo Wang*) [issue #2] 36
  - protrusion patch `eqnum` for `IEEEtran` 35
- Font aliases: declare CMU Serif as an alias of New Computer Modern 143
- `\MT@apply@patch`: compatibility with `babel/spanish`: fix catcodes 34
- `\MT@get@prot`: compatibility fix for `csquotes` (reported by *Gustavo Barros*) [issue #1] 52
  - reset counters 52
  - set penalties to zero and vfuzz to max (reported by *florian*) 52
- `\MT@prot@addgroup`: rescan grouped material to allow `verbatim` etc. (reported by *Torsten Schuetze*) 54
- `\MT@redefine@patch`: facilitate redefinition of patches 33
- `\MT@with@babel@and@T`: fix grouping 28

2021/12/10

Version 3.0b

- General: disable patches for tex4ht 31
- \MT@get@prot: unconditionally \leavevmode 52
- \MT@prot@addgroup: don't rescan anymore (reported by *Akira Yokosawa* and *theufman*) [issues #3,#4] 54
- \MT@prot@l@: make \long again (reported by *Akira Yokosawa*) [issue #3] 52

2022/02/22

Version 3.0c

- General: info that protrusion patch eqnum may not be effective with mathtools (reported by *user182849*) 36
  - make babel option work with polyglossia 139
  - protrusion patch footnote: also for minipages 36
- \MT@csq@eqgroup: compatibility with csquotes, again (reported by *NightShade*) 53
- \MT@noindent: use \RawNoindent, if available [issue #8] 53
- \MT@prot@iffirstcmd: use \long variant of \@car (reported by *frafl*) [issue #6] 54
- \MT@prot@l@: next try at removing \long 52
- \MT@prot@r: no longer \long 52
- \noprotrusionifhmode: new command 52

2022/03/14

Version 3.0d

- \lslig: define \font@name (reported by *Ulrike Fischer*) [issue #12] 72
- \MT@get@prot: no longer reset counters (reported by *Brian Dunn*) [issue #14] 53
- \MT@prot@addgroup: no brace delimiting (reported by *Nelson Lago*) [issue #10] 54

2022/06/20

Version 3.0e

- General: disable csquotes's grouping control for \leftprotrusion (reported by *Ralf Steinle* and *Denis Bitouzé*) 31
  - fix protrusion patch footnote for hyperref if \hyper@nopatch@footnote is defined 36
  - fix protrusion patch footnote for hyperref with option implicit=false 36
  - fix protrusion patch footnote with beamer 37
  - protrusion patch eqnum: adjust to robustness of \eqref (LaTeX 2022-06-01), cf. [issue #19] 36
- Font aliases: declare aliases for stickstoo fonts 143
- \MT@csq@eqgroup: add \relax (for csquotes) 53
- \MT@get@prot: revert \leavevmode 52
- \MT@is@active@hook: fix for doc/shortvrb's \MakeShortVerb 90
  - fix for listings's \lstMakeShortInline (reported by *Denis Bitouzé*) 90
  - hook for active chars 90
- \MT@prot@addgroup: only add group if very first token 54
- \MT@prot@check@cmds: add standard font selection commands 59
  - ignore \@empty 59
  - new macro: make list of commands extensible 59
- \MT@prot@get@first@token: don't gobble previously captured content 55
- \MT@prot@ifmacro@: use etoolbox facilities 54

2022/06/23

Version 3.0f

- \MT@prot@check@cmds: add new NFSS commands 59
  - add microtype's commands 59
- \MT@prot@iflicrcmd: no need to know about encodings (also fixes [issue #20], reported by *Christophe Dervieux*) 54

2023/03/06

Version 3.1

- General: fix test for KOMA classes in protrusion patch footnote (also reported by *David Purton*) [issue #26] 37
  - new protrusion patch verbatim 37
- Documentation: add hint about protrusion patch warnings U29
- \DeclareMicrotypeFilePrefix: new command (suggested by *rallg*) [issue #28] 107
- \MT@active@features: always activate contexts for tracking 97
- \MT@define@patch: check if patch defined 33
  - define patches globally 33
- \MT@find@file: simplify 81
- \MT@prot@addgroup: allow multiple groups 54
- \MT@prot@check: allow replacement command 56
- \MT@prot@check@cmds: add ltxdoc and doc abbreviations 59
  - add \em 59
  - add \MakeUppercase and friends 59
  - add csquotes's commands (reported by *Shen Zhou Hong*) [issue #25] 60
  - add fontaxes commands 59
  - add nfssext-cfr commands 60
- \MT@prot@check@E: new macro: take care of commands that enclose their argument 57
- \MT@prot@check@e: new macro: take care of starred commands that enclose their argument 57
- \MT@prot@check@eX: new macro: replace enclosing commands 57
- \MT@prot@check@F: new macro: for text commands 58

\MT@prot@check@l: new macro: for commands that enclose their second argument 57

\MT@prot@check@lX: new macro: replace language-switching enclosing commands 58

\MT@prot@get@first@token: expand toks once 55

\MT@prot@l@tc: new macro: for text commands 52

2023/03/13 Version 3.1a

\MT@prot@check@F: fix: execute (and empty) \MT@toks (reported by *Artur A. Marczok* and *Uwe Siart*) 58

2024/03/29 Version 3.1b

General: add DEPENDS.txt (suggested by *Bruno Victal*) U1

\MT@check@MT@version: check if .sty and .def (and .lua) file versions match 125

\MT@fix@fontdimen@six: compatibility with LuaTeX in DVI mode (reported by *Linas Stonys*) 40

\MT@if@expanding@F: new macro: check whether inside expanding context 20

\MT@prot@check@C: new macro: for text commands with an optional argument (after *scholnik* reported that \MakeUppercase and friends support one) [issue #29] 58

\MT@prot@check@F: fix: failed when group began with another command [issue #31] 58

\MT@with@babel@and@T: don't use \@pkgextension (was \@onlypreambled in older LaTeX versions) 28

\pickup@font: guard against expanding context 96

2024/12/12 Version 3.2

General: fix protrusion patch footnote for tagging (reported by *hpvd*) [issue #40] 36

letterspacing/tracking with XeTeX (requested by *Aleksandr Petrosyan*) [issue #22] 66

protrusion patch eqnum: fix for IEEEtran 36

protrusion patch eqnum: fix for showkeys 35

Tracking: features instead of no ligatures with XeTeX 151

\DeclareMicrotypeBabelHook: make catcode-safe 108

\microtypesetup: make robust 127

\MT@get@prot: make color-safe 53

\MT@get@size: check for dimen (reported by *Oliver Beery*) [issue #36] 103

\MT@if@opentype@font: version for XeTeX 25

\MT@letterspace@default: new default for letterspacing: 50 (following a report by *Mark Collins*) [issue #41] 17

\MT@ls@fontspec@font: fix for spaces in font names, made necessary because fontspec v2.9b no longer strips them (reported by *Bernhard Fisseni*) [issue #35] 70

\MT@prot@addtoken@next: make color-safe 56

\MT@prot@check@cmds: add color's \color command 59

add some fontspec commands 60

\MT@prot@check@F: fix: add \@empty (reported by *nowox*) 58

\MT@prot@check@F@beamer: compatibility with beamer's overlay specifications (reported by *Frank Mittelbach*) [issue #38] 58

\MT@prot@check@o: new macro: for one-argument commands with optional argument 57

\MT@set@tr@codes: fix noligatures for OpenType fonts with letterspace 68

\MT@tr@noligatures: new values 'none' and 'all' for the 'no ligatures' key in \SetTracking (after a report by *user202729*) 73

\MT@tr@set@okern: fix for LuaTeX and XeTeX 78

\MT@tr@set@space@@: fix for LuaTeX and XeTeX 75

\MT@varwidth@setup: patch varwidth to allow for margin kerning (reported by '*web-stranger*') 29

\SetTracking: new key 'features' to en-/disable fontspec features (LuaTeX or XeTeX) 109

D Index

Numbers in upright shape refer to the *page* where the corresponding entry is described (bold face) resp. occurs. Numbers in italic shape refer to the *code line* where the corresponding entry is defined (underlined) resp. used. Numbers prefixed with 'U' refer to the User manual.

Options

E The LATEX Project Public License

LPPL Version 1.3c 2008-05-04

Copyright 1999, 2002–2008 LATEX3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

Preamble

The LATEX Project Public License (LPPL) is the primary license under which the LATEX kernel and the base LATEX packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is TEX-related (such as a LATEX package), but it is written in such a way that you can use it even if your work is unrelated to TEX.

The section 'WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE', below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the LATEX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document '`cfgguide.tex`' and '`modguide.tex`' in the base LATEX distribution for suggestions.

Definitions

In this license document the following terms are used:

Work: Any work being distributed under this License.

Derived Work: Any work that under any applicable law is derived from the Work.

Modification: Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify: To apply any procedure that produces a Derived Work under any applicable law.

Distribution: Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun's Network File System (NFS).

Compiled Work: A version of the Work that has been processed into a form where it is directly usable on a computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer: A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the 'Copyright Holder' under any applicable law.

Base Interpreter: A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a 'LATEX-Format' or in the case of files belonging to the 'LATEX-format' a program implementing the 'TEX language'.

Conditions on Distribution and Modification

1. Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.

2. You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.

3. You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.

4. If you are the Current Maintainer of the Work, you may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.

5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.

6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.

   (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.

   (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.

   (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.

   (d) You distribute at least one of the following with the Derived Work:

      i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;

      ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.

7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.

8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.

9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.

10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.

    (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.

11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.

12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work 'as is', without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status 'author-maintained' if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is 'author-maintained'.

The Work has the status 'maintained' if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status 'maintained' to 'unmaintained' if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.

2. If this search is successful, then enquire whether the Work is still maintained.

   (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.

   (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a LaTeX work, this could be done, for example, by posting to `comp.text.tex`.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.

   (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.

4. If you make an 'intention announcement' as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.

5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as 'you' in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document '`modguide.tex`' in the base LATEX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing LATEX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to LATEX, the discussion in '`modguide.tex`' may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
%% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%   https://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status `maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the 'Work' referring to the three files '`pig.dtx`', '`pig.ins`', and '`pig.sty`' (the last being generated from '`pig.dtx`' using '`pig.ins`'), the 'Base Interpreter' referring to any 'LATEX-Format', and both 'Copyright Holder' and 'Current Maintainer' referring to the person 'M. Y. Name'.

If you do not want the Maintenance section of LPPL to apply to your Work, change 'maintained' above into 'author-maintained'. However, we recommend that you use 'maintained' as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

Important Recommendations

Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
 % This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.