

The fonttable package

Author: Peter Wilson, Herries Press

Maintainer: L^AT_EX Team

<https://github.com/LaTeX-Package-Repositories/herries-press/issues>

v1.6e 2026-04-19

Abstract

The package lets you typeset the characters in a font in tabular and/or running text forms.

Contents

1	Introduction	1
2	The package	1
2.1	Table and texts	2
2.2	Testing a glyph	3
3	The code	4
3.1	Table and texts	4
3.2	Testing a glyph	12
	Bibliography	14

1 Introduction

The fonttable package lets you typeset a font’s character set in tabular and/or running text forms.

This manual is typeset according to the conventions of the L^AT_EX DOC-STRIP utility which enables the automatic extraction of the L^AT_EX macro source files [MG04].

2 The package

The package provides commands to typeset a table of all the glyphs in a given font and to typeset an example of regular text. For font designers it provides commands to typeset a ‘test’ glyph among sets of glyphs from the font.

`\fnthours` As a convenience, `\fnthours` prints the time of day when the file was processed; it uses the 24 hour clock notation. (The macro `\today` prints the date when the file was processed.)

2.1 Table and texts

`\fonttable` The command

```
\fonttable{<testfont>}
```

typesets a table showing all the glyphs in the `<testfont>`, where `<testfont>` is the name of a font file¹ like `cmr10` (for Computer Modern Roman) or `pzdr` (for Zapf Dingbats).

NOTE: The `mftinc` package [Pak05] for pretty-printing METAFONT code also defines a `\fonttable` macro that is akin to this one. If you want to use both packages together then you can use the following general procedure for when a macro `\macro` is defined in both `packA` and `packB` packages.

```
\usepackage{packA}
\let\macroA\macro%    save packA's definition
\let\macro\relax%     undefine \macro
\usepackage{packB}%   now it's packB's definition of \macro
...
\macro %    use the packB definition
\macroA %   use the packA definition
```

`\xfonttable` The command

```
\xfonttable{<encoding>}{<family>}{<series>}{<shape>}
```

typesets a table showing all the glyphs in the font with encoding `<encoding>` (e.g., T1 or OMS), family `<family>` (e.g., `ppl` for Palatino or `cmbrs` for CM Bright Math (OMS)), font series `<series>` (e.g., `sb` for semibold of `m` for medium), and font shape `<shape>` (e.g., `n` for normal or `sc` for small caps). For example:

```
\xfonttable{U}{pzd}{m}{n}
```

for Zapf Dingbats.

`\pikfont` The command²

```
\pikfont{<encoding>}{<family>}{<series>}{<shape>}
```

selects the font with encoding `<encoding>` (e.g., T1 or OMS), family `<family>` (e.g., `ppl` for Palatino or `cmbrs` for CM Bright Math (OMS)), font series `<series>` (e.g., `sb` for semibold of `m` for medium), and font shape `<shape>` (e.g., `n` for normal or `sc` for small caps). For example:

```
\pikfont{T1}{ppl}{m}{sc}
```

for Palatino small caps. The size of the font corresponds to the current setting (e.g., `\footnotesize`, `\normalsize`, `\Large`). It can also be changed after being selected by the incantation

```
\fontsize{<size>}{<baselineskip>}\selectfont
```

where `<size>` is the normal height and `<baselineskip>` is the distance between text lines; the measurement system is `pts` but just use numbers with no units specified. For example:

```
\fontsize{12}{15}\selectfont
```

for a 12pt font with 15pts between baselines.

If you are unsure about the meaning of the various arguments of `\xfonttable` and `\pikfont` see *The Companion* [MG04, Chapter 7] or the *LaTeX2e font selection* manual (`fntguide.tex`; try `texdoc fntguide`).

`\fontrange` The package attempts to populate the table with a maximum of 256 glyphs,

¹More precisely, the name of a `.tfm` file.

²The name was chosen in an attempt to avoid clashes with other macros that might perform similar functions.

numbered from 0 to 255. The `\fontrange{⟨low⟩}{⟨high⟩}` declaration changes this by reducing the range so that it extends from `⟨low⟩` to `⟨high⟩`, where `⟨low⟩` should be at least 0 and `⟨high⟩` at most 256, and `⟨low⟩` less than `⟨high⟩`.

The table is composed of blocks of sixteen characters. If necessary the value of `⟨low⟩` is adjusted lower and `⟨high⟩` is adjusted higher to match this block structure. For example, if you wanted a table of the lower 128 characters then `\fontrange{0}{127}` would do the job, while the upper half of a 256 character font could be tabulated via `\fontrange{128}{255}`.

`\decimals` Normally each cell in the table includes the decimal number of the position in the (256) character set. `\nodecimals` turns off this numbering and `\decimals` turns it on. The default is `\decimals`.

`\hexoct` Normally the columns and rows in the table are numbered using hexadecimal and octal numbers. These can be turned off by `\nohexoct` and turned on again with `\hexoct`, which is the default.

`\fablewidth` The font table's width is the length `\fablewidth`, which by default is set to the normal textwidth (or more exactly, to `\hsize`). The table itself is left aligned. However, if `\nohexoct` is in effect the width of the table is its natural width.

`\fntcolwidth` When `\nohexoct` is in effect the minimum width of a table column is `\fntcolwidth`. This is initially declared as `\setwidth{\fntcolwidth}{0.08\fablewidth}`

`\fonttext` The command `\fonttext{⟨testfont⟩}` typesets an example text using the `⟨testfont⟩` (e.g. `cmr10`).

`\simpletext` The example text can be just a paragraph and a line of capitals, or include `\fulltext` more complex accented words as well. Following the declaration `\fulltext` the complex words are included as well as the example paragraph. The default is `\simpletext` for just the paragraph.

`\regulartext` The command `\regulartext{⟨fontspec⟩}` typesets the example text using `⟨fontspec⟩`, for example `\rmfamily\itshape` or `\pikfont{T1}{pnc}{m}{it}`.

`\fonttexts` The macro `\fonttexts{⟨testfont⟩}{⟨text⟩}` typesets `⟨text⟩` using the `⟨testfont⟩`

`\regulartexts` (e.g., `cmr10`). Similarly the macro `\regulartexts{⟨fontspec⟩}{⟨text⟩}` typesets `⟨text⟩` using `⟨fontspec⟩` (e.g., `\rmfamily\itshape` or `\pikfont{T1}{ppl}{m}{it}`).

`\germanparatext` `\germanparatext` expands to a German language paragraph, borrowed from the `blindtext` package [Lik05]. `\latinparatext` expands to one version of a paragraph of the traditional *lorem ipsum* dummy Latin text. Either, or both, of these could be used as the `⟨text⟩` argument to `\fonttexts` or `\regulartexts`.

NOTE: These were originally called `\germantext` and `\latintext` but on 2009/05/14 I was told that the `babel` package defines `\latintext`, which causes unexpected results if it is used in the same document as this package. To try and be on the safe side I renamed `\germantext` as well as `\latintext`.

`\aztext` `\aztext` expands to the lowercase Latin alphabet a to z, and `\AZtext` is the

`\AZtext` corresponding command for the uppercase A to Z. The macros `\digitstext` and

`\digitstext` `\puncttext` expand respectively to the digits 0 to 9, and to the typical punctuation

`\puncttext` marks. In all cases there is a space between each character.

2.2 Testing a glyph

The macros here are a reimplementaion of Donald Knuth's `testfont.tex`, which is available from CTAN.

In the following, the value of a glyph argument can be specified as its location in the font (i.e., as a decimal number). With a few exceptions, if the glyph is

The glyphs are taken from the current font. If the font does not have Latin alphabet glyphs in the ASCII locations then in the descriptions below phrases like ‘lowercase alphabet’ or ‘uppercase alphabet’ or ‘digits’, should be taken to mean (the glyphs in) those locations.

3 The code

3.1 Table and texts

³Sometimes called a ‘backquote’.

`\sevenrm` A small fixed size roman font.

```
2 \providecommand*\sevenrm{\fontsize{7}{9pt}\rmfamily}
```

`\f@tm` Counts and a dimen.

```
\f@tn 3 \newcount\f@tm \newcount\f@tn \newcount\f@tp \newdimen\f@tdim
```

```
\f@tp 4
```

`\f@tdim`

`\fonttable` `\fonttable{}` typesets a table of all the glyphs in the `` (e.g., aunc10).

```
5 \newcommand*\fonttable}[1]{%
```

```
6   \def\f@tfontname{#1}%
```

```
7   \bgroup
```

```
8   \f@tstartfont
```

```
9   \f@table
```

```
10  \egroup}
```

```
11
```

`\pikfont` `\pikfont{<encoding>}{<family>}{<series>}{<shape>}` selects the font with `<encoding>`, `<family>`, `<series>` and `<shape>`.

```
12 \DeclareRobustCommand*\pikfont}[4]{%
```

```
13   \fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}\selectfont}
```

```
14
```

`\xfonttable` `\xfonttable{<encoding>}{<family>}{<series>}{<shape>}` typesets a table of all the glyphs in the font with `<encoding>`, `<family>`, `<series>` and `<shape>` (e.g., `\xfonttable{T1}{pnc}{m}{it}` for New Century Schoolbook italic). The original code for the macro was supplied by Enrico Gregorio.

```
15 \newcommand*\xfonttable}[4]{%
```

```
16   \begingroup
```

```
17   \pikfont{#1}{#2}{#3}{#4}%
```

```
18   \edef\f@tfontname{\fontname\font}%
```

New: strip any size information from the fontname (which could be, e.g., either ‘cmr10’ or ‘cmr10_{at}10pt’.) This wasn’t necessary before because we didn’t explicitly choose the font size; it was inferred automatically.

```
19   \edef\@tempa{ \string a\string t}%
```

```
20   \edef\@tempb{\noexpand\in@\@tempa{\f@tfontname}}%
```

```
21   \@tempb
```

```
22   \ifin@
```

```
23     \edef\f@tfontname{\expandafter\f@tstripsize\f@tfontname}%
```

```
24   \fi
```

End new code, and finish as before:

```
25   \normalfont
```

```
26   \f@tstartfont
```

```
27   \f@table
```

```
28   \endgroup
```

```
29 }
```

`\f@tstripsize` Needed above.

```
30 \edef\@tempa{%
```

```
31   \def\noexpand\f@tstripsize
```

```
32     ##1 \string a\string t##2\string p\string t{##1}%
```

```
33 }
```

```
34 \@tempa
```

`\f@tstartfont` Sets up for a font table.

```
35 \newcommand*{\f@tstartfont}{%
```

New: scale the font by 0.01% to (attempt to) avoid TeX's font optimisation. This becomes a problem in Spanish babel, say, when `\textfont\fam` changes when `cmr10` has been loaded under a different name, here. (And the `\textfont` can no longer be parsed correctly. See: <http://latex-alive.tumblr.com/post/3229118083/texs-font-loading-optimisation>)

```
36 \tempdima=\f@size pt
```

```
37 \font\f@ttestfont=\f@tfontname\space at 0.9999\tempdima\relax
```

Continue as before:

```
38 \f@ttestfont \f@tsetbaselineskip
```

```
39 \ifdim\fontdimen6\f@ttestfont<10pt\relax
```

```
40 \rightskip=0pt plus 20pt\relax
```

```
41 \else
```

```
42 \rightskip=0pt plus 2em\relax
```

```
43 \fi
```

```
44 \spaceskip=\fontdimen2\f@ttestfont % space between words (\raggedright)
```

```
45 \xspaceskip=\fontdimen2\f@ttestfont
```

```
46 \advance\xspaceskip by\fontdimen7\f@ttestfont
```

```
47 }
```

`\f@tsetbaselineskip`

```
48 \newcommand*{\f@tsetbaselineskip}{\setbox0=\hbox{\f@tn=0
```

```
49 \loop\char\f@tn \ifnum \f@tn<255 \advance\f@tn 1 \repeat}
```

```
50 \baselineskip=6pt \advance\baselineskip\ht0 \advance\baselineskip\dp0 }
```

```
51
```

`\f@toct` `\f@toct{<onum>}` typesets the octal constant `<onum>`.

```
52 \newcommand*{\f@toct}[1]{\hbox{\rmfamily\'{}\kern-.2em\itshape
```

```
53 #1\kern.05em}} % octal constant
```

`\f@thex` `\f@thex{<hnum>}` typesets the hexadecimal constant `<hnum>`.

```
54 \newcommand*{\f@thex}[1]{\hbox{\rmfamilyH{}\ttfamily#1}} % hexadecimal constant
```

`\f@tsetdigs` `\f@tsetdigs`

```
55 \def\f@tsetdigs#1"#2{\gdef\f@thexpre{#2}% \f@thexpre=hex prefix; \f@tocta\f@toctb=corre
```

```
56 \f@tm=\f@tn \divide\f@tm by 64 \xdef\f@tocta{\the\f@tm}%
```

```
57 \multiply\f@tm by-64 \advance\f@tm by\f@tn \divide\f@tm by 8 \xdef\f@toctb{\the\f@tm}}
```

`\f@ttestrow` `\f@ttestrow` checks if there are any characters in the next block of 16 slots.

```
58 \newcommand*{\f@ttestrow}{\setbox0=\hbox{\penalty 1\def\\{\char"\f@thexpre}%
```

```
59 \\0\\1\\2\\3\\4\\5\\6\\7\\8\\9\\A\\B\\C\\D\\E\\F%
```

```
60 \global\f@tp=\lastpenalty}} % \f@tp=1 if none of the characters exist
```

```
61
```

`\ifhexoct` Flag for (not) setting hex and octal numbers.

```
\hexoct 62 \newif\ifhexoct
```

```
\nohexoct 63 \newcommand*{\hexoct}{\hexocttrue}
```

```
64 \newcommand*{\nohexoct}{\hexoctfalse}
```

```
65 \hexoct
```

```
66
```

```

\fontoddlinenum \fontoddline
67 \newcommand*{\fontoddline}{\cr
68 \noalign{\nointerlineskip}
69 \multispan{19}\hrulefill&
70 \setbox0=\hbox{\lower 2.3pt\hbox{\fontthe{\fontthe x}}}\smash{\box0}
71 \cr
72 \noalign{\nointerlineskip}}
73

```

```

\iff@tskipping
\fontskippingtrue 74 \newif\iff@tskipping
\fontskippingfalse 75

```

`\fontrange` `\fontrange{<low>}{<high>}` sets the character range to be output.

```

76 \newcommand*{\fontrange}[2]{%
77 \ifnum#1<#2\relax

```

Set `\fontlow` to the nearest multiple of 16 that is at or below `<low>`, but first make sure that it will be at least 0.

```

78 \ifnum#1<\z@
79 \fontm=\z@
80 \else
81 \fontm=#1
82 \divide \fontm \sixt@@n
83 \multiply \fontm \sixt@@n
84 \fi
85 \edef\fontlow{\the\fontm}

```

Set `\fonthigh` to the nearest multiple of 16 at or above `<high>`, finally making sure that its maximum is 256.

```

86 \fontm=#2
87 \divide \fontm \sixt@@n
88 \advance \fontm \@ne
89 \multiply \fontm \sixt@@n
90 \ifnum \fontm > \ccclvi \fontm=\ccclvi \fi
91 \edef\fonthigh{\the\fontm}
92 \else
93 \PackageError{fonttable}{%
94 Improper values for fontrange. Default values substituted}{\@ehc}
95 \def\fontlow{0} \def\fonthigh{256}
96 \fi}
97 \fontrange{0}{256}
98

```

`\fontloopforsixteen` `\fontloopforsixteen` sets up a block of sixteen character slots.

```

99 \newcommand*{\fontloopforsixteen}{%
100 \ifnum\fontn<\fontlow \global\fontn=\fontlow\fi
101 \loop\fontskippingfalse
102 \ifnum\fontn<\fonthigh \fontm=\fontn \divide\fontm \sixt@@n \chardef\next=\fontm
103 \expandafter\fontsetdigs\meaning\next \fonttestrow
104 \ifnum\fontp=\@ne \fontskippingtrue \fi\fi
105 \iff@tskipping \global\advance\fontn \sixt@@n \repeat}
106

```

`\f@tevenlinenonum` does something similar when no external numbers are printed.

`\f@tmorechart` `\f@tmorechart` sets two lines of the table, and `\f@tmorechartnonum` does the `\f@tmorechartnonum` same when there are no external numbers.

`\f@tchartline` `\f@tchartline` does a line of the table, including external numbers, and
`\f@tsimpleline` `\f@tsimpleline` does an unnumbered line.

`\f@tchartstrut` `\f@tchartstrut` is a strut used in each table line. `\f@twidth` is width of an `\f@twidth` externally numbered table. `\f@tcolwidth` is the minimum width of a column in `\f@tcolwidth` an unnumbered table.

```

132 \newcommand*{\f@tchartstrut}{\lower4.5pt\ vbox to14pt{}}
133 \newdimen\ftablewidth
134 \ftablewidth=\hsize
135 \newdimen\fntcolwidth
136 \setlength{\fntcolwidth}{0.08\ftablewidth}

```

```

\ftcol \ftstartchartnonum is a table line of spaces, with no verticals.
\ftstartchartnonum 137 \newcommand*{\ftcol}{%
138 \multicolumn{1}{c}{\hspace*{\fntcolwidth}}%
139 \newcommand*{\ftstartchartnonum}{%
140 \ftcol &\ftcol &\ftcol &\ftcol &\ftcol &\ftcol &\ftcol &\ftcol}
141
```


`\f@tftablenum` `\f@tftablenum` sets a complete character table. The actual code is in either `\f@tftablenum` or `\f@tftablenonum` for externally numbered or plain tables, respectively.

```

142 \newcommand*{\f@tftablenum}{\global\f@tn=\z@
143   \halign to\f@tablewidth\bgroup
144     \f@tchartstrut##\tabskip0pt plus10pt&
145     &\hfil##\hfil&\vrule##\cr
146     \lower6.5pt\null
147     &&&\f@toct0&&\f@toct1&&\f@toct2&&\f@toct3&&\f@toct4&&\f@toct5&&\f@toct6&&\f@toct7&%
148     \f@tevenline}
149 \newcommand*{\f@tftablenonum}{%
150   \global\f@tn=\z@
151   \begin{tabular}{|c|c|c|c|c|c|c|c|}
152     \f@tstartchartnonum
153     \f@tevenlinenonum
154   \end{tabular}}
155 \newcommand*{\f@tftable}{\ifhexoct\f@tftablenum\else\f@tftablenonum\fi}
156

```

`\f@tendchart` `\f@tendchart` sets the last line of an externally numbered table with the relevant hex digits.

```

157 \newcommand*{\f@tendchart}{\cr\noalign{\hrule}
158   \raise11.5pt\null&&&\f@thex 8&&\f@thex 9&&\f@thex A&&\f@thex B&
159   &\f@thex C&&\f@thex D&&\f@thex E&&\f@thex F&\cr
160   \egroup$$\par}
161

```

`\f@tpsg` `\f@tpsg` typesets a single glyph, possibly with its decimal slot number. `\f@placechar` is the function to typeset the glyph with its number that is internally defined as `\f@placedecimal` if decimals are to be shown.

```

162 \newcommand*{\f@tpsg}{%
163   \setbox\z@=\hbox{\f@placechar{\char\f@tn}{\the\f@tn}}%
164   \ifdim\ht\z@>7.5pt\relax
165     \f@treposition
166   \else
167     \ifdim\dp\z@>2.5pt\relax
168       \f@treposition
169     \fi
170   \fi
171   \box\z@
172   \global\advance\f@tn\@ne
173 }

```

Change this definition to adjust the typesetting of the decimal numbers:

```

174 \newcommand*\f@placedecimal[2]{#1\ {\tiny #2}}

```

`\decimals` Following `\decimals`, which is the default, decimal numbers are printed in the table. Following `\nodecimals` they are not printed.

```

175 \newcommand*{\nodecimals}{%
176   \renewcommand*\f@placechar{\@firstoftwo}%
177 }
178 \newcommand{\decimals}{%
179   \renewcommand*\f@placechar{\f@placedecimal}%
180 }

```

```

181 \newcommand*\f@placechar{}
182 \decimals

\f@treposition \f@treposition
183 \newcommand*\f@treposition{\setbox0=\vbox{\kern2pt\box0}\f@tdim=\dp0
184 \advance\f@tdim 2pt \dp0=\f@tdim}
185

\fonttext \fonttext{<font>} typesets \knutext using <font> (e.g. aunc110).
186 \def\fonttext#1{%
187   \def\f@tfontname{#1}%
188   \bgroup
189   \f@tstartfont
190   \knutext
191   \egroup}
192

\regulartext \regulartext{<fontspec>} typesets \knutext using <fontspec> (e.g., \aunc1family).
193 \def\regulartext#1{%
194   \bgroup
195   #1
196   \knutext
197   \egroup}
198

\knutext Deathless prose from Knuth for testing a font. It includes \moreknutext,
\capknutext, and \knunames.
199 \def\knutext{{
200 On November 14, 1885, Senator \& Mrs.~Leland Stanford called together
201 at their San Francisco mansion the 24~prominent men who had been
202 chosen as the first trustees of The Leland Stanford Junior University.
203 They handed to the board the Founding Grant of the University, which
204 they had executed three days before. This document---with various
205 amendments, legislative acts, and court decrees---remains as the
206 University's charter. In bold, sweeping language it stipulates that
207 the objectives of the University are "to qualify students for
208 personal success and direct usefulness in life; and to promote the
209 publick welfare by exercising an influence in behalf of humanity and
210 civilization, teaching the blessings of liberty regulated by law, and
211 inculcating love and reverence for the great principles of government
212 as derived from the inalienable rights of man to life, liberty, and
213 the pursuit of happiness."
214
215 \moreknutext
216
217 \capknutext
218
219 \knunames
220 \par}}
221

\@moreknutext Some more text with a variety of ligatures and accents.
222 \def\@moreknutext{'But aren't Kafka's Schlo{\ss} and {\AE}sop's

```

```

223 {\OE}uvres often na{"\i}ve vis-\'a-vis the d{\ae}monic ph{\oe}nix's
224 official r\^ole in fluffy souffl'es? }
225

```

`\@capkntext` Text using only capital letters and some punctuation.

```

\capkntext 226 \newcommand{\@capkntext}{%
227 (!'THE DAZED BROWN FOX QUICKLY GAVE 12345--67890 JUMPS!)}
228 \let\capkntext\@capkntext
229

```

`\@knunames` Lots of accents masquerading in personal names.

```

230 \def\@knunames{ {\AA}ngel\aa\ Beatrice Claire
231   Diana \Erica Fran\c{c}oise Ginette H\'el\'ene Iris
232   Jackie K\=aren {\L}au\ra Mar{"\i}a N\H{a}taf{\l}{\u{i}}e {\O}ctave
233   Pauline Qu\^eneau Roxanne Sabine T\~a{"'\j}a Ur\v{s}ula
234   Vivian Wendy Xanthippe Yv{\o}nne Z\'azilie\par}
235

```

`\guillemotleft` Just in case the French quotes are not defined, as they are called for in the subse-
`\guillemotright` quent `\germantext`.

```

\flqq 236 \DeclareTextSymbol{\guillemotleft}{OT1}{\'{\'}}
\frqq 237 \DeclareTextSymbol{\guillemotright}{OT1}{\'{\'}}
238 \providecommand{\flqq}{\guillemotleft}
239 \providecommand{\frqq}{\guillemotright}
240

```

`\germantext` Text from the Blindtext package.

```

\germanparatext 241 \providecommand*\@germantext{%
242   \PackageWarning{fonttable}{\protect\germantext\space is deprecated,
243     \MessageBreak use \protect\germanparatext\space instead}}
244 \newcommand*\@germanparatext{%
245 Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer
246 diesen Text liest, ist selbst schuld. Der Text gibt lediglich den
247 Grauwert der Schrift an. Ist das wirklich so? Ist es
248 gleich-g"ul-tig ob ich schreibe: \frqq Dies ist ein
249 Blindtext\flqq\ oder \frqq Huardest gefburn\flqq? Kjift --
250 mitnichten! Ein Blindtext bietet mir wichtige Informationen. An
251 ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie
252 harmonisch die Figuren zueinander stehen und pr"u-fe, wie breit
253 oder schmal sie l"aufft. Ein Blindtext sollte m"og\~lichst viele
254 verschiedene Buchstaben enthalten und in der Originalsprache
255 gesetzt sein. Er mu\ss\ keinen Sinn ergeben, sollte aber lesbar
256 sein. Fremdsprachige Texte wie \frqq Lorem ipsum\flqq\ dienen
257 nicht dem eigentlichen Zweck, da sie eine
258 falsche Anmutung vermitteln.\par}
259

```

`\latintext` The traditional printers' text.

```

\latinparatext 260 \providecommand*\@latintext{%
261   \PackageWarning{fonttable}{\protect\latintext\space may be overridden by the
262     babel package \MessageBreak use
263     \protect\latinparatext\space instead}}
264 \newcommand*\@latinparatext{%

```

```

265 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam
266 lobortis facilisis sem. Nullam nec mi et neque pharetra
267 sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper,
268 felis non sodales commodo, lectus velit ultrices augue, a
269 dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie
270 ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in
271 sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
272 Duis fringilla tristique neque. Sed interdum libero ut metus.
273 Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit
274 amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
275 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a,
276 turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum
277 turpis accumsan semper.\par}
278

\simpletext \simpletext kills off \moreknuttext and \knunames. \fulltext restores \moreknuttext
\fulltext and \knunames. Make \fulltext the default.
\moreknuttext 279 \newcommand*{\simpletext}{\let\moreknuttext\relax \let\knunames\relax}
\knunames 280 \newcommand*{\fulltext}{\let\moreknuttext\@moreknuttext \let\knunames\@knunames}
281 \fulltext
282

fonttexts \fonttexts{<font>}{<text>} typesets <text> using <font> (e.g. aunc110).
283 \def\fonttexts#1#2{%
284 \def\fontname{#1}%
285 \bgroup
286 \fontstartfont
287 #2
288 \egroup}
289

\regulartexts \regulartext{<fontspec>}{<text>} typesets <text> using <fontspec> (e.g., \aunc1family).
290 \def\regulartexts#1#2{%
291 \bgroup
292 #1 #2
293 \egroup}
294

\aztext The various characters used for Latin texts.
\AZtext 295 \newcommand*{\aztext}{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}
\digitstext 296 \newcommand*{\AZtext}{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}
\punctext 297 \newcommand*{\digitstext}{0 1 2 3 4 5 6 7 8 9}
298 \newcommand*{\punctext}{' ! @ $ % & * ( ) \_ - + = [ ] < > \{ \} : ; ' , . ? /}
299

```

3.2 Testing a glyph

This is a reimplementaion of Donald Knuth's `testfont.tex` which is available from CTAN and there is also a commented version in Appendix H of *The META-FONT Book*.

```

\fnthours The time of day on a 24 hour clock.
\f@ttwodigits 300 %%% using \tempcnta for Knuth's \m and \tempcntb for his \n

```

```

301 \newcommand*{\fnthours}{\@tempcntb=\time \divide\@tempcntb 60
302   \@tempcnta=-\@tempcntb \multiply\@tempcnta 60 \advance\@tempcnta \time
303   \f@ttwodigits\@tempcntb:\f@ttwodigits\@tempcnta}
304 \newcommand*{\f@ttwodigits}[1]{\ifnum #1<10 0\fi \number#1}
305

```

`\f@tgettsechars` `\f@tgettsechars{⟨T⟩}{⟨S⟩}{⟨E⟩}` gets three characters and `\chardefs` `\f@ttchar` `\f@ttchar` to `⟨T⟩` (the test character), `\f@tschar` to `⟨S⟩` (start character) and `\f@techar` to `\f@tschar` `⟨E⟩` (the end character).

```

\f@techar 306 \newcommand*{\f@tgettsechars}[3]{%
307   \chardef\f@ttchar=#1 \chardef\f@tschar=#2 \chardef\f@techar=#3}
308

```

`\glyphmixture` `\glyphmixture{⟨T⟩}{⟨S⟩}{⟨E⟩}` sets a mix of `⟨T⟩` within the glyph range from `\f@tmixpattern` `⟨S⟩` to `⟨E⟩` according to the pattern `\f@tmixpattern`. The work is done by `\f@tdomix` `\f@tdomix`.

```

309 \newcommand*{\glyphmixture}[3]{\f@tgettsechars{#1}{#2}{#3}%
310   \f@tdomix\f@tmixpattern}
311 \newcommand*{\f@tmixpattern}{\f@tocta\f@toctb\f@tocta\f@tocta\f@toctb\f@toctb\f@tocta\f@tocta\f@to
312 \newcommand*{\f@tdomix}[1]{\par\chardef\f@tocta=\f@ttchar \f@tempcntb=\f@tschar
313   \loop \chardef\f@toctb=\@tempcntb #1\endgraf
314   \ifnum \@tempcntb<\f@techar \advance\@tempcntb \@ne \repeat}
315

```

`\glyphalternation` These are similar to `\glyphmixture` and `\f@tmixpattern` except that the glyphs `\f@taltpattern` are alternated.

```

316 \newcommand*{\glyphalternation}[3]{\f@tgettsechars{#1}{#2}{#3}%
317   \f@tdomix\f@taltpattern}
318 \newcommand*{\f@taltpattern}{\f@tocta\f@toctb\f@tocta\f@toctb\f@tocta\f@toctb\f@tocta\f@toctb\f@to
319

```

`\f@tdisc` For breaking long lines so that the test character will be at the end of one line and repeated at the start of the next one.

```

320 \newcommand*{\f@tdisc}{\discretionary{\f@ttchar}{\f@ttchar}{\f@ttchar}}
321

```

`\glyphseries` `\glyphseries{⟨T⟩}{⟨S⟩}{⟨E⟩}` puts the test character `⟨T⟩` between all the others `\f@tdoseries` in the range `⟨S⟩` to `⟨E⟩`. The work is done by `\f@tdoseries`.

```

322 \newcommand*{\glyphseries}[3]{\f@tgettsechars{#1}{#2}{#3}%
323   \f@tdisc\f@tdoseries\f@tschar\f@techar\par}
324 \newcommand*{\f@tdoseries}[2]{\@tempcntb=#1\relax
325   \loop\char\@tempcntb\f@tdisc
326   \ifnum\@tempcntb<#2\advance\@tempcntb \@ne \repeat}
327

```

`\glyphalphabet` `\glyphalphabet{⟨T⟩}` inserts the test glyph `⟨T⟩` between the lowercase alpha-
`\GLYPHALPHABET` betic characters. Similarly `\GLYPHALPHABET{⟨T⟩}` does the same with the up-
`\f@tcomplower` percase characters. The work is done by, respectively, `\f@tcomplower` and
`\f@tcompupper` `\f@tcompupper`.

```

328 \newcommand*{\glyphalphabet}{\f@tcomplower}
329 \newcommand*{\GLYPHALPHABET}{\f@tcompupper}
330 \newcommand*{\f@tcomplower}[1]{\chardef\f@ttchar=#1

```

```

331 \f@tdisc\f@tdoseries{'a'}{'z'}\f@tdoseries{31}{34}\par}
332 \newcommand*{\f@tcompupper}[1]{\chardef\f@ttchar=#1
333 \f@tdisc\f@tdoseries{'A'}{'Z'}\f@tdoseries{35}{37}\par}
334
\glyphlowers These macros generate an extended mix of characters of a particular kind. The
\glyphhppers work is done by \f@tdocomprehensive with \f@tcllc, \f@tcuc, and \f@tdgs setting
\glyphdigits up the glyph sets.
    \f@tcllc 335 \newcommand*{\glyphlowers}{\f@tdocomprehensive\f@tcllc{'a'}{'z'}{31}{34}}
    \f@tcuc 336 \newcommand*{\glyphhppers}{\f@tdocomprehensive\f@tcuc{'A'}{'Z'}{35}{37}}
    \f@tdgs 337 \newcommand*{\glyphdigits}{\f@tdocomprehensive\f@tdgs{'0'}{'4'}{'5'}{'9'}}
\f@tdocomprehensive 338 \newcommand*{\f@tdocomprehensive}[5]{\par\chardef\f@ttchar=#2
339 \loop{#1} \ifnum\f@ttchar<#3\@tempcnta=\f@ttchar\advance\@tempcnta \@ne
340 \chardef\f@ttchar=\@tempcnta \repeat
341 \chardef\f@ttchar=#4
342 \loop{#1} \ifnum\f@ttchar<#5\@tempcnta=\f@ttchar\advance\@tempcnta \@ne
343 \chardef\f@ttchar=\@tempcnta \repeat}
344 \newcommand*{\f@tcllc}{\f@tdisc\f@tdoseries{'a'}{'z'}\f@tdoseries{31}{34}\par}
345 \newcommand*{\f@tcuc}{\f@tdisc\f@tdoseries{'A'}{'Z'}\f@tdoseries{35}{37}\par}
346 \newcommand*{\f@tdgs}{\f@tdisc\f@tdoseries{'0'}{'9'}\par}
347

\glyphpunct \glyphpunct sets punctuation marks in combination with different sorts of letters.
\f@tdopunct The work is done by \f@tdopunct.
348 \newcommand*{\glyphpunct}{\par\f@tdopunct{min}\f@tdopunct{pig}\f@tdopunct{hid}
349 \f@tdopunct{HIE}\f@tdopunct{TIP}\f@tdopunct{fluff}}
350 \$1,234.56 + 7/8 = 9\% @ \#0\par}
351 \newcommand*{\f@tdopunct}[1]{#1,\ #1:\ #1;\ #'1'\
352 ?'#1?\ !'#1!\ (#1)\ [#1]\ #1*\ #1.\par}
353

The end of the package.
354 \pack}

```

Bibliography

- [Lik05] Knut Lickert. *Blindtext.sty: Creating text for testing / Texterzeugung zum testen*. October 2005. (Available from CTAN in [macros/latex/contrib/blindtext](#))
- [MG04] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Second edition. Addison-Wesley Publishing Company, 2004.
- [Pak05] Scott Pakin ‘The mftinc package’, January 2005. (Available from CTAN in [macros/latex/contrib/mftinc](#))

Index

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

Symbols

" [223](#), [234](#), [248](#), [252](#), [253](#)

- \# 350
 \\$ 298, 350
 \% 350
 \& 200, 298
 \- 248, 252, 253
 \/ 53
 \= 232
 \@capknuttext 226
 \@ccclvi 90
 \@ehc 94
 \@firstoftwo 176
 \@knunames ... 230, 280
 \@moreknuttext . 222, 280
 \@tempa .. 19, 20, 30, 34
 \@tempb 20, 21
 \@tempcntb
 . 300, 301, 302,
 303, 312, 313,
 314, 324, 325, 326
 \{ 298
 \} 298
 _ 351
 \^ 224, 233
 _ 298
 \' 223, 231, 237
 \~ 233
- A**
- \AA 230
 \aa 230
 \AE 222
 \ae 223
 \AZtext 3, 295
 \aztext 3, 295
- B**
- \baselineskip 50
- C**
- \c 231
 \capknuttext ... 217, 226
 \char ... 49, 58, 163, 325
 \chardef 102, 307, 312,
 313, 330, 332,
 338, 340, 341, 343
- D**
- \decimals 3, 175
 \DeclareRobustCommand
 12
 \DeclareTextSymbol .
 236, 237
 \digitstext 3, 295
 \discretionary 320
 \dp ... 50, 167, 183, 184
- E**
- \edef 18, 19,
 20, 23, 30, 85, 91
 \endgraf 313
- F**
- \f@placechar
 . 162, 176, 179, 181
 \f@placedecimal 162, 179
 \f@size 36
 \f@taltpattern 316
 \f@tchartline
 121, 122, 127
 \f@tchartstrut
 130, 132, 144
 \f@tcllc 335
 \f@tcol 137
 \f@tcomplower 328
 \f@tcompupper 328
 \f@tcuc 335
 \f@tdgs 335
 \f@tdim 3, 183, 184
 \f@tdisc 320,
 323, 325, 331,
 333, 344, 345, 346
 \f@tdocomprehensive 335
 \f@tdomix 309, 317
 \f@tdopunct 348
 \f@tdoseries 322, 331,
 333, 344, 345, 346
 \f@techar . 306, 314, 323
 \f@tendchart .. 109, 157
 \f@tevenline
 107, 122, 148
 \f@tevenlinenonum .
 107, 125, 153
 \f@tfontname . 6, 18,
 20, 23, 37, 187, 284
 \f@tftablenonum ... 142
 \f@tftablenum 142
 \f@tgettsechars ...
 . 306, 309, 316, 322
 \f@thex . 54, 70, 158, 159
 \f@thexpre .. 55, 58, 70
 \f@thigh
 91, 95, 102, 109, 113
 \f@tloopforsixteen .
 99, 108, 112
 \f@tlow 85, 95, 100
 \f@tm 3, 56, 57,
 79, 81, 82, 83,
 85, 86, 87, 88,
 89, 90, 91, 102, 121
 \f@tmixpattern 309
- \f@tmorechart . 109, 120
 \f@tmorechartnonum .
 117, 120
 \f@tn 3, 48, 49,
 56, 57, 100, 102,
 105, 109, 113,
 142, 150, 163, 172
 \f@toct ... 52, 128, 147
 \f@tocta 55, 56,
 128, 311, 312, 318
 \f@toctb 55, 57, 121,
 128, 311, 313, 318
 \f@todddline 67, 121
 \f@todddlinenum 67
 \f@tp 3, 60, 104
 \f@tpsg ... 128, 130, 162
 \f@treposition
 165, 168, 183
 \f@tschar . 306, 312, 323
 \f@tsetbaselineskip
 38, 48
 \f@tsetdigs 55, 103
 \f@tsimpleline
 124, 125, 127
 \f@tskippingfalse .
 74, 101
 \f@tskippingtrue 74, 104
 \f@tstartchartnonum
 137, 152
 \f@tstartfont
 8, 26, 35, 189, 286
 \f@tstripsize ... 23, 30
 \f@ttchar 306,
 312, 320, 330,
 332, 338, 339,
 340, 341, 342, 343
 \f@ttestfont ... 37,
 38, 39, 44, 45, 46
 \f@ttestrow 58, 103
 \f@ttwodigits 300
 \flqq 236, 249, 256
 \fntcolwidth 3, 132, 138
 \fnthours 1, 300
 \font 18, 37
 \fontdimen 39, 44, 45, 46
 \fontencoding 13
 \fontfamily 13
 \fontname 18
 \fontrange 2, 76
 \fontseries 13
 \fontshape 13
 \fonttable 2, 5
 \fonttext 3, 186
 \fonttexts . 3, 283, 283

- \frqq . 236, 248, 249, 256
- \ftable 9, 27, 142
- \ftablewidth 3, 132, 143
- \fulltext 3, 279
- G**
- \germanparatext . 3, 241
- \germantext 241
- \GLYPHALPHABET . . 4, 328
- \glyphalphabet . . 4, 328
- \glyphalternation 4, 316
- \glyphdigits . . . 4, 335
- \glyphlower . . 4, 4, 335
- \glyphmixture . . . 4, 309
- \glyphpunct 4, 348
- \glyphseries 4, 322
- \glyphuppers 335
- \guillemotleft . . . 236
- \guillemotright . . . 236
- H**
- \H 54, 232
- \hexoct 3, 62
- \hexoctfalse 64
- \hexocttrue 63
- \hline 114, 116, 124
- \hrulefill 69
- \ht 50, 164
- I**
- \i 223, 232
- \iff@tskipping . 74, 105
- \ifhexoct 62, 155
- \ifin@ 22
- \in@ 20
- \itshape 52
- J**
- \j 233
- K**
- \knunames 219, 279
- \knutext . . 190, 196, 199
- L**
- \L 232
- \l 232
- \lastpenalty 60
- \latinparatext . . 3, 260
- \latintext 260
- \loop 49, 101, 313, 325, 339, 342
- \lower 70, 132, 146
- M**
- \m 300
- \meaning 103
- \MessageBreak . 243, 262
- \moreknutext . . 215, 279
- \multicolumn 138
- \multiply 57, 83, 89, 302
- \multispan 69
- N**
- \n 300
- \newcount 3
- \newif 62, 74
- \next . 102, 103, 109, 110
- \noalign 68, 72, 120, 157
- \nodecimals 3, 175
- \noexpand 20, 31
- \nohexoct 3, 62
- \nointerlineskip 68, 72
- \normalfont 25
- O**
- \O 232
- \o 234
- \OE 223
- \oe 223
- P**
- \PackageError 93
- \PackageWarning 242, 261
- \pikfont 2, 12, 17
- \providecommand . 2, 238, 239, 241, 260
- \puncttext 3, 295
- R**
- \raise 158
- \regulartext 3, 193
- \regulartexts . . . 3, 290
- \repeat 49, 105, 314, 326, 340, 343
- S**
- \sevenrm 2
- \simpletext 3, 279
- \sixt@@n 82, 83, 87, 89, 102, 105
- \smash 70
- \spaceskip 44
- \ss 222, 255
- \string 19, 32
- T**
- \tabskip 144
- \time 301, 302
- \tiny 174
- \ttfamily 54
- U**
- \u 232
- V**
- \v 233
- \vbox 132, 183
- \vrule 145
- X**
- \xdef 56, 57, 121
- \xfonttable 2, 15
- \xspaceskip 45, 46