

The fonttable package*

Author: Peter Wilson, Herries Press

Maintainer: Will Robertson

will dot robertson at latex-project dot org

2009/09/22

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	2
2.1	Table and texts	2
2.2	Testing a glyph	4
3	The code	5
3.1	Table and texts	5
3.2	Testing a glyph	13
	Bibliography	15

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].

*This file has version number v1.5d, last revised 2009/09/22.

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:

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.

`\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, 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 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>` (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 `\latinparatext`

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 $\langle text \rangle$ 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 corresponding command for the uppercase A to Z. The macros `\digitstext` and `\puncttext` expand respectively to the digits 0 to 9, and to the typical punctuation 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 within the visible ASCII range (33–126) it may instead be specified by the ASCII character prefixed with a single open quote mark³ (`'`). The exceptions are nos: 37 (`%`), 92 (`\`) 123 (`{`) and 125 (`}`) (but there may be others). In any case, the glyph representing the character `p` can be specified either as `'p` or as 112.

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.

`\glyphmixture` `\glyphmixture{<T>}{<S>}{<E>}` typesets the $\langle T \rangle$ (test) glyph between the glyphs in the range from $\langle S \rangle$ (start) to $\langle E \rangle$ (end). For example `\glyphmixture{'e'}{'f'}{'g'}` will produce
`efeeffeefffef`
`egeeggeeeegggeg`

`\glyphalternation` `\glyphalternation{<T>}{<S>}{<E>}` typesets the $\langle T \rangle$ glyph alternately between each glyph in the range from $\langle S \rangle$ to $\langle E \rangle$. For example `\glyphalternation{'e'}{'f'}{'g'}` will produce
`efefefefefefefefef`
`egegegegegegegege`

`\glyphseries` `\glyphseries{<T>}{<S>}{<E>}` typesets the $\langle T \rangle$ glyph between the glyphs in the range from $\langle S \rangle$ to $\langle E \rangle$. For example `\glyphseries{'e'}{'f'}{'h'}` will produce
`efegehe`

`\glyphalphabet` `\glyphalphabet{<T>}` typesets the $\langle T \rangle$ glyph between each letter of the lowercase Latin alphabet plus a few others. `\GLYPHALPHABET{<T>}` does the same but using the uppercase Latin alphabet. For example, the output of

³Sometimes called a 'backquote'.

`\glyphalphabet`{‘3} is like
`3a3b3c3d3e3f3g...3z303~3!3"3`

`\glyphlowers` `\glyphlowers` takes each character of the lowercase alphabet in turn as a test
`\glyphlowers` glyph and sets it interspersed among the other lowercase characters. `\glyphhuppers`
`\glyphdigits` and `\glyphdigits` are similar except that they use the uppercase alphabet and
the ten digits instead. For example, `\glyphdigits` produces output like
000102030405060708090
101112131415161718191
202122232425262728292
...
909192939495969798999

`\glyphpunct` `\glyphpunct` sets a collection of words with an assortment of punctuation
marks.

3 The code

```

1 (*pack)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{fonttable}[2009/09/22 v1.5d displays a font]
4

```

3.1 Table and texts

Most of the code below is an edited version of code used in `nfssfont.tex` for displaying aspects of the set of glyphs in a font.

```

\sevenrm     A small fixed size roman font.
             5 \providecommand*\sevenrm{\fontsize{7}{9pt}\rmfamily}

\fontm     Counts and a dimen.
\fontn     6 \newcount\fontm \newcount\fontn \newcount\fontp \newdimen\fontdim
\fontp     7
\fontdim     \fonttdim
\fonttable     \fonttable{<font>} typesets a table of all the glyphs in the <font> (e.g., aunc10).
             8 \newcommand*\fonttable[1]{%
             9     \def\fontname{#1}%
             10     \bgroup
             11     \fontstartfont
             12     \ftable
             13     \egroup}
             14

\pikfont     \pikfont{<encoding>}{<family>}{<series>}{<shape>} selects the font with <encoding>,
             <family>, <series> and <shape>.
             15 \DeclareRobustCommand\pikfont[4]{%
             16     \fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}\selectfont}
             17

```

```

\fonttable \fonttable{⟨encoding⟩}{⟨family⟩}{⟨series⟩}{⟨shape⟩} typesets a table of all
the glyphs in the font with ⟨encoding⟩, ⟨family⟩, ⟨series⟩ and ⟨shape⟩ (e.g.,
\fonttable{T1}{pnc}{m}{it} for New Century Schoolbook italic). The original
code for the macro was supplied by Enrico Gregorio.
18 \newcommand*{\fonttable}[4]{\bgroup
19 \pikfont{#1}{#2}{#3}{#4}%
20 \edef\fontname{\fontname\font}\normalfont
21 \fontstartfont
22 \ftable
23 \egroup}
24

\fontstartfont Sets up for a font table.
25 \newcommand*{\fontstartfont}{\font\fonttestfont=\fontname
26 \fonttestfont \fontsetbaselineskip
27 \ifdim\fontdimen6\fonttestfont<10pt \rightskip=0pt plus 20pt
28 \else\rightskip=0pt plus 2em \fi
29 \spaceskip=\fontdimen2\fonttestfont % space between words (\raggedright)
30 \xspaceskip=\fontdimen2\fonttestfont \advance\xspaceskip
31 by\fontdimen7\fonttestfont}
32

\fontsetbaselineskip
33 \newcommand*{\fontsetbaselineskip}{\setbox0=\hbox{\font=0
34 \loop\char\font \ifnum \font<255 \advance\font 1 \repeat}
35 \baselineskip=6pt \advance\baselineskip\ht0 \advance\baselineskip\dp0 }
36

\fonttoct \fonttoct{⟨onum⟩} typesets the octal constant ⟨onum⟩.
37 \newcommand*{\fonttoct}[1]{\hbox{\rmfamily\'}{\kern-.2em\itshape
38 #1/\kern.05em}} % octal constant

\fontthex \fontthex{⟨hnum⟩} typesets the hexadecimal constant ⟨hnum⟩.
39 \newcommand*{\fontthex}[1]{\hbox{\rmfamily\H{}\ttfamily#1}} % hexadecimal constant

\fontsetdigs \fontsetdigs
40 \def\fontsetdigs#1"#2{\gdef\h{#2}% \h=hex prefix; \0\1=corresponding octal
41 \fontm=\font \divide\fontm by 64 \xdef\0{\the\fontm}%
42 \multiply\fontm by-64 \advance\fontm by\font \divide\fontm by 8 \xdef\1{\the\fontm}}

\fonttestrow \fonttestrow checks if there are any characters in the next block of 16 slots.
43 \newcommand*{\fonttestrow}{\setbox0=\hbox{\penalty 1\def\{\char"\h}%
44 \0\1\2\3\4\5\6\7\8\9\A\B\C\D\E\F%
45 \global\fontp=\lastpenalty}} % \fontp=1 if none of the characters exist
46

\ifhexoct Flag for (not) setting hex and octal numbers.
\hexoct 47 \newif\ifhexoct
\nohexoct

```

```

48 \newcommand*{\hexoct}{\hexocttrue}
49 \newcommand*{\nohexoct}{\hexoctfalse}
50 \hexoct
51

\iff@toddlinenum \f@todddline
52 \newcommand*{\f@todddline}{\cr
53 \noalign{\nointerlineskip}
54 \multispan{19}\hrulefill&
55 \setbox0=\hbox{\lower 2.3pt\hbox{\f@thex{\h x}}}\smash{\box0}
56 \cr
57 \noalign{\nointerlineskip}}
58

\iff@tskipping
\f@tskippingtrue 59 \newif\iff@tskipping
\f@tskippingfalse 60

\fonrange \fonrange{<low>}{<high>} sets the character range to be output.
61 \newcommand*{\fonrange}[2]{%
62 \ifnum#1<#2\relax

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

63 \ifnum#1<\z@
64 \f@tm=\z@
65 \else
66 \f@tm=#1
67 \divide \f@tm \sixt@@n
68 \multiply \f@tm \sixt@@n
69 \fi
70 \edef\f@tlow{\the\f@tm}

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

71 \f@tm=#2
72 \divide \f@tm \sixt@@n
73 \advance \f@tm \@ne
74 \multiply \f@tm \sixt@@n
75 \ifnum \f@tm > \@ccclvi \f@tm=\@ccclvi \fi
76 \edef\f@thigh{\the\f@tm}
77 \else
78 \PackageError{fonttable}{%
79 Improper values for fonrange. Default values substituted}{\@ehc}
80 \def\f@tlow{0} \def\f@thigh{256}
81 \fi}
82 \fonrange{0}{256}
83

\f@tloopforsixteen \f@tloopforsixteen sets up a block of sixteen character slots.

```

<code>\f@tevenline</code>	<code>\f@tevenline</code> gets next non-empty set of a block of 16 characters. It either calls
<code>\f@tevenlinenonum</code>	<code>\f@tmorechart</code> to print them, or <code>\f@tendchart</code> to finish off the table if all 256 potential characters have been processed.

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

<code>\f@tchartline</code>	<code>\f@tchartline</code> does a line of the table, including external numbers, and
<code>\f@tsimpleline</code>	<code>\f@tsimpleline</code> does an unnumbered line.

<code>\f@tchartstrut</code>	<code>\f@tchartstrut</code> is a strut used in each table line. <code>\f@tchartwidth</code> is width of an
<code>\f@tchartwidth</code>	externally numbered table. <code>\f@tchartcolwidth</code> is the minimum width of a column in
<code>\f@tchartcolwidth</code>	an unnumbered table.


```

117 \newcommand*{\f@tchartstrut}{\lower4.5pt\vbox to14pt{}}
118 \newdimen\ftablewidth
119 \ftablewidth=\hsize
120 \newdimen\fntcolwidth
121 \setlength{\fntcolwidth}{0.08\ftablewidth}

```

`\f@tcol` `\f@tstartchartnonum` is a table line of spaces, with no verticals.

```

\f@tstartchartnonum 122 \newcommand*{\f@tcol}{%
123 \multicolumn{1}{c}{\hspace*{\fntcolwidth}}}
124 \newcommand*{\f@tstartchartnonum}{%
125 \f@tcol &\f@tcol &\f@tcol &\f@tcol &\f@tcol &\f@tcol &\f@tcol &\f@tcol}
126

```

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

```

\f@tftablenonum 127 \newcommand*{\f@tftablenum}{\global\f@tn=\z@
128 \halign to\ftablewidth\bgroup
129 \f@tchartstrut#\tabskip0pt plus10pt&
130 &\hfil#\hfil&\vrule#\cr
131 \lower6.5pt\null
132 &&&\f@toct0&&\f@toct1&&\f@toct2&&\f@toct3&&\f@toct4&&\f@toct5&&\f@toct6&&\f@toct7&%
133 \f@tevenline}
134 \newcommand*{\f@tftablenonum}{%
135 \global\f@tn=\z@
136 \begin{tabular}{|c|c|c|c|c|c|c|c|}
137 \f@tstartchartnonum
138 \f@tevenlinenonum
139 \end{tabular}}
140 \newcommand*{\ftable}{\ifhexoct\f@tftablenum\else\f@tftablenonum\fi}
141

```

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

```

142 \newcommand*{\f@tendchart}{\cr\noalign{\hrule}
143 \raise11.5pt\null&&&\f@thex 8&&\f@thex 9&&\f@thex A&&\f@thex B&
144 &\f@thex C&&\f@thex D&&\f@thex E&&\f@thex F&\cr
145 \egroup$$\par}
146

```

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

`\f@tpsg` `\f@tpsg` typesets a single glyph, possibly with its decimal slot number.

NOTE (2009/04/30): Initially `\f@tpsg` was called `\:`, which LaTeX defines to be a medium space. No doubt this was OK with the interactive version but can cause havoc when used in a package!

```

147 \newcommand*\f@tpsg{}
148 \newcommand*\nodelcimals{%
149 \renewcommand*\f@tpsg{%
150 \setbox0=\hbox{\char\f@tn}%

```

```

151     \ifdim\ht0>7.5pt\fontreposition
152     \else\ifdim\dp0>2.5pt\fontreposition\fi\fi
153     \box0\global\advance\fontn 1 %
154 }%
155 }

156 \newcommand{\decimals}{%
157   \renewcommand*\fontpsg{%
158     \setbox0=\hbox{\char\fontn\, \rlap{\tiny \the\fontn}}%
159     \ifdim\ht0>7.5pt\fontreposition
160     \else\ifdim\dp0>2.5pt\fontreposition\fi\fi
161     \box0\global\advance\fontn 1 %
162   }%
163 }
164 \decimals

\fontreposition \fontreposition
165 \newcommand*\fontreposition{\setbox0=\vbox{\kern2pt\box0}\fontdim=\dp0
166   \advance\fontdim 2pt \dp0=\fontdim}
167

\fonttext \fonttext{<font>} typesets \knutext using <font> (e.g. aunc10).
168 \def\fonttext#1{%
169   \def\fontname{#1}%
170   \bgroup
171   \fontstartfont
172   \knutext
173   \egroup}
174

\regulartext \regulartext{<fontspec>} typesets \knutext using <fontspec> (e.g., \aunc1family).
175 \def\regulartext#1{%
176   \bgroup
177   #1
178   \knutext
179   \egroup}
180

\knutext Deathless prose from Knuth for testing a font. It includes \moreknutext,
\capknutext, and \knunames.
181 \def\knutext{
182 On November 14, 1885, Senator \& Mrs.~Leland Stanford called together
183 at their San Francisco mansion the 24~prominent men who had been
184 chosen as the first trustees of The Leland Stanford Junior University.
185 They handed to the board the Founding Grant of the University, which
186 they had executed three days before. This document---with various
187 amendments, legislative acts, and court decrees---remains as the
188 University's charter. In bold, sweeping language it stipulates that
189 the objectives of the University are 'to qualify students for
190 personal success and direct usefulness in life; and to promote the

```

191 publick welfare by exercising an influence in behalf of humanity and
 192 civilization, teaching the blessings of liberty regulated by law, and
 193 inculcating love and reverence for the great principles of government
 194 as derived from the inalienable rights of man to life, liberty, and
 195 the pursuit of happiness.’’

196

197 \moreknutext

198

199 \capknutext

200

201 \knunames

202 \par}}

203

\@moreknutext Some more text with a variety of ligatures and accents.

204 \def\@moreknutext{?'But aren't Kafka's Schlo{\ss} and {\AE}sop's
 205 {\OE}uvres often na{"i}ve vis-\'a-vis the d{\ae}monic ph{\oe}nix's
 206 official r\`ole in fluffy souffl\'es? }

207

\@capknutext Text using only capital letters and some punctutation.

\capknutext 208 \newcommand{\@capknutext}{%
 209 (!'THE DAZED BROWN FOX QUICKLY GAVE 12345--67890 JUMPS!)}
 210 \let\capknutext\@capknutext

211

\@knunames Lots of accents masquerading in personal names.

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

217

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

\flqq 218 \DeclareTextSymbol{\guillemotleft}{OT1}{\'{\'}}
 \frqq 219 \DeclareTextSymbol{\guillemotright}{OT1}{\'{\'}}
 220 \providecommand{\flqq}{\guillemotleft}
 221 \providecommand{\frqq}{\guillemotright}

222

\germantext Text from the Blindtext package.

\germanparatext 223 \providecommand*\germantext{%
 224 \PackageWarning{fonttable}{\protect\germantext\space is deprecated,
 225 \MessageBreak use \protect\germanparatext\space instead}}
 226 \newcommand*\germanparatext{%
 227 Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer
 228 diesen Text liest, ist selbst schuld. Der Text gibt lediglich den

```

229 Grauwert der Schrift an. Ist das wirklich so? Ist es
230 gleich\-g\"ul\"-tig ob ich schreibe: \frqq Dies ist ein
231 Blindtext\flqq\ oder \frqq Huardest gefburn\flqq? Kjift --
232 mitnichten! Ein Blindtext bietet mir wichtige Informationen. An
233 ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie
234 harmonisch die Figuren zueinander stehen und pr\"u\"-fe, wie breit
235 oder schmal sie l\"auft. Ein Blindtext sollte m\"og\"-lichst viele
236 verschiedene Buchstaben enthalten und in der Originalsprache
237 gesetzt sein. Er mu\"ss\ keinen Sinn ergeben, sollte aber lesbar
238 sein. Fremdsprachige Texte wie \frqq Lorem ipsum\flqq\ dienen
239 nicht dem eigentlichen Zweck, da sie eine
240 falsche Anmutung vermitteln.\par}
241

\latintext The traditional printers' text.
\latinparatext 242 \providecommand*{\latintext}{%
243   \PackageWarning{fonttable}{\protect\latintext\space may be overridden by the
244     babel package \MessageBreak use
245       \protect\latinparatext\space instead}}
246 \newcommand*{\latinparatext}{%
247 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam
248 lobortis facilisis sem. Nullam nec mi et neque pharetra
249 sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper,
250 felis non sodales commodo, lectus velit ultrices augue, a
251 dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie
252 ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in
253 sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
254 Duis fringilla tristique neque. Sed interdum libero ut metus.
255 Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit
256 amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
257 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a,
258 turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum
259 turpis accumsan semper.\par}
260

\simpletext \simpletext kills off \morekntext and \knunames. \fulltext restores \morekntext
\fulltext and \knunames. Make \fulltext the default.
\morekntext 261 \newcommand*{\simpletext}{\let\morekntext\relax \let\knunames\relax}
\knunames 262 \newcommand*{\fulltext}{\let\morekntext\@morekntext \let\knunames\@knunames}
263 \fulltext
264

fonttexts \fonttexts{<font>}{<text>} typesets <text> using <font> (e.g. auncl10).
265 \def\fonttexts#1#2{%
266   \def\f@tfontname{#1}%
267   \bgroup
268   \f@tstartfont
269   #2
270   \egroup}
271

```

```

\regulartexts \regulartext{<fontspec>}{<text>} typesets <text> using <fontspec> (e.g., \aunclfamily).
272 \def\regulartexts#1#2{%
273   \bgroup
274   #1 #2
275   \egroup}
276

\aztext The various characters used for Latin texts.
\AZtext 277 \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 278 \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}
\puncttext 279 \newcommand*{\digitstext}{0 1 2 3 4 5 6 7 8 9}
280 \newcommand*{\puncttext}{' ! @ $ % & * ( ) _ - + = [ ] < > \{ \} : ; ' , . ? /}
281

```

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.
\fttwodigits 282 %%%%%%%%% using \@tempcnta for Knuth's \m and \@tempcntb for his \n
283 \newcommand*{\fnthours}{\@tempcntb=\time \divide \@tempcntb 60
284   \@tempcnta=-\@tempcntb \multiply \@tempcnta 60 \advance \@tempcnta \time
285   \f@ttwodigits \@tempcntb:\f@ttwodigits \@tempcnta}
286 \newcommand*{\f@ttwodigits}[1]{\ifnum #1<10 0\fi \number#1}
287

\ftgettsechars \ftgettsechars{<T>}{<S>}{<E>} gets three characters and \chardefs \ftttchar
\ftttchar to <T> (the test character), \f@tschar to <S> (start character) and \f@techar to
\f@tschar <E> (the end character).
\fttechar 288 \newcommand*{\ftgettsechars}[3]{%
289   \chardef \ftttchar=#1 \chardef \f@tschar=#2 \chardef \f@techar=#3}
290

\glyphmixture \glyphmixture{<T>}{<S>}{<E>} sets a mix of <T> within the glyph range from
\ftmixpattern <S> to <E> according to the pattern \ftmixpattern. The work is done by
\ftdomix \ftdomix.
291 \newcommand*{\glyphmixture}[3]{\ftgettsechars{#1}{#2}{#3}%
292   \ftdomix \ftmixpattern}
293 \newcommand*{\ftmixpattern}{\0\1\0\0\1\1\0\0\0\1\1\1\0\1}
294 \newcommand*{\ftdomix}[1]{\par \chardef \0=\ftttchar \@tempcntb=\f@tschar
295   \loop \chardef \1=\@tempcntb #1\endgraf
296   \ifnum \@tempcntb<\f@techar \advance \@tempcntb \@ne \repeat}
297

\glyphalternation These are similar to \glyphmixture and \ftmixpattern except that the glyphs
\ftaltpattern are alternated.

```

```

298 \newcommand*\glyphalternation}[3]{\f@tgettsechars{#1}{#2}{#3}%
299                                     \f@tdomix\f@taltpattern}
300 \newcommand*\f@taltpattern}{\0\1\0\1\0\1\0\1\0\1\0\1\0\1\0\1\0}
301

\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.
302 \newcommand*\f@tdisc{\discretionary{\f@ttchar}{\f@ttchar}{\f@ttchar}}
303

\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.
304 \newcommand*\glyphseries}[3]{\f@tgettsechars{#1}{#2}{#3}%
305   \f@tdisc\f@tdoseries\f@tschar\f@techar\par}
306 \newcommand*\f@tdoseries}[2]{\@tempcntb=#1\relax
307   \loop\char\@tempcntb\f@tdisc
308     \ifnum\@tempcntb<#2\advance\@tempcntb \@ne \repeat}
309

\glyphalphabet \glyphalphabet{<T>} inserts the test glyph <T> between the lowercase alpha-
\GLYPHALPHABET betic characters. Similarly \GLYPHALPHABET{<T>} does the same with the up-
percase characters. The work is done by, respectively, \f@tcomplower and
\f@tcompupper \f@tcompupper.
310 \newcommand*\glyphalphabet{\f@tcomplower}
311 \newcommand*\GLYPHALPHABET{\f@tcompupper}
312 \newcommand*\f@tcomplower}[1]{\chardef\f@ttchar=#1
313   \f@tdisc\f@tdoseries{‘a’}{‘z’}\f@tdoseries{31}{34}\par}
314 \newcommand*\f@tcompupper}[1]{\chardef\f@ttchar=#1
315   \f@tdisc\f@tdoseries{‘A’}{‘Z’}\f@tdoseries{35}{37}\par}
316

\glyphlowers These macros generate an extended mix of characters of a particular kind. The
\glyphuppers work is done by \f@tdocomprehensive with \f@tcllc, \f@tcuc, and \f@tdgs setting
\glyphdigits up the glyph sets.
317 \newcommand*\glyphlowers{\f@tdocomprehensive\f@tcllc{‘a’}{‘z’}{31}{34}}
318 \newcommand*\glyphuppers{\f@tdocomprehensive\f@tcuc{‘A’}{‘Z’}{35}{37}}
319 \newcommand*\glyphdigits{\f@tdocomprehensive\f@tdgs{‘0’}{‘4’}{‘5’}{‘9’}}
320 \f@tdocomprehensive \newcommand*\f@tdocomprehensive}[5]{\par\chardef\f@ttchar=#2
321   \loop{#1} \ifnum\f@ttchar<#3\@tempcnta=\f@ttchar\advance\@tempcnta \@ne
322   \chardef\f@ttchar=\@tempcnta \repeat
323   \chardef\f@ttchar=#4
324   \loop{#1} \ifnum\f@ttchar<#5\@tempcnta=\f@ttchar\advance\@tempcnta \@ne
325   \chardef\f@ttchar=\@tempcnta \repeat}
326 \newcommand*\f@tcllc{\f@tdisc\f@tdoseries{‘a’}{‘z’}\f@tdoseries{31}{34}\par}
327 \newcommand*\f@tcuc{\f@tdisc\f@tdoseries{‘A’}{‘Z’}\f@tdoseries{35}{37}\par}
328 \newcommand*\f@tdgs{\f@tdisc\f@tdoseries{‘0’}{‘9’}\par}
329

```

```

\glyphpunct \glyphpunct sets punctuation marks in combination with different sorts of letters.
\ftdopunct  The work is done by \ftdopunct.

330 \newcommand*{\glyphpunct}{\par\ftdopunct{min}\ftdopunct{pig}\ftdopunct{hid}
331                               \ftdopunct{HIE}\ftdopunct{TIP}\ftdopunct{fluff}}
332   \$1,234.56 + 7/8 = 9\% @ \#0\par}
333 \newcommand*{\ftdopunct}[1]{#1,\ #1:\ #1;\ '#1'\
334   ?'#1?\ !'#1!\ (#1)\ [#1]\ #1*\ #1.\par}
335

The end of the package.

336 </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			
\"	205, 216, 230, 234, 235	\{	280
\#	332	\}	280
\\$	280, 332	\^	206, 215
\%	332	_	280
\&	182, 280	\‘	205, 213, 219
\,	158	\~	215
\-	230, 234, 235	Numbers	
\/	38	\0	40, 41, 113, 293, 294, 300
\=	214	\1	40, 42, 106, 113, 293, 295, 300
\@capknuttext	<u>208</u>	A	
\@cclvi	75	\AA	212
\@ehc	79	\aa	212
\@knunames	212, 262	\AE	204
\@moreknuttext	204, 262	\ae	205
\@tempcntb	282–285, 294–296, 306–308	\AZtext	4, <u>277</u>
		\aztext	4, <u>277</u>
		B	
		\baselineskip	35
		C	
		\c	213
		\capknuttext	199, <u>208</u>
		\char	34, 43, 150, 158, 307
		\chardef	87, 289, 294, 295, 312, 314,

- 320, 322, 323, 325
- D**
- \decimals 3, 147
- \DeclareRobustCommand
..... 15
- \DeclareTextSymbol .
..... 218, 219
- \digitstext 4, 277
- \discretionary 302
- \dp 35, 152, 160, 165, 166
- E**
- \edef 20, 70, 76
- \endgraf 295
- F**
- \f@taltpattern 298
- \f@tchartline
.... 106, 107, 112
- \f@tchartstrut
.... 115, 117, 129
- \f@tclrc 317
- \f@tcol 122
- \f@tcomplower 310
- \f@tcompupper 310
- \f@tcuc 317
- \f@tdgs 317
- \f@tdim 6, 165, 166
- \f@tdisc 302, 305, 307,
313, 315, 326–328
- \f@tdocomprehensive 317
- \f@tdomix 291, 299
- \f@tdopunct 330
- \f@tdoseries .. 304,
313, 315, 326–328
- \f@tchar . 288, 296, 305
- \f@tendchart ... 94, 142
- \f@tevenline 92, 107, 133
- \f@tevenlinenonum .
..... 92, 110, 138
- \f@tfontname
9, 20, 25, 169, 266
- \f@tftablenonum ... 127
- \f@tftablenum 127
- \f@tgettsechars ...
. 288, 291, 298, 304
- \f@thex . 39, 55, 143, 144
- \f@thigh 76, 80, 87, 94, 98
- \f@tloopforsixteen .
..... 84, 93, 97
- \f@tlow 70, 80, 85
- \f@tm 6, 41, 42, 64, 66–
68, 70–76, 87, 106
- \f@tmixpattern 291
- \f@tmorechart .. 94, 105
- \f@tmorechartnonum .
..... 102, 105
- \f@tn .. 6, 33, 34, 41,
42, 85, 87, 90,
94, 98, 127, 135,
150, 153, 158, 161
- \f@toct ... 37, 113, 132
- \f@toddlne 52, 106
- \f@toddlneum 52
- \f@tp 6, 45, 89
- \f@tpsg ... 113, 115, 147
- \f@treposition 151,
152, 159, 160, 165
- \f@tschar . 288, 294, 305
- \f@tsetbaselineskip
..... 26, 33
- \f@tsetdigs 40, 88
- \f@tsimpleline
.... 109, 110, 112
- \f@tskippingfalse 59, 86
- \f@tskippingtrue 59, 89
- \f@tstartchartnonum
..... 122, 137
- \f@tstartfont
11, 21, 25, 171, 268
- \f@ttchar
. 288, 294, 302,
312, 314, 320–325
- \f@ttestfont
.... 25–27, 29–31
- \f@ttestrow 43, 88
- \f@ttwodigits 282
- \flqq 218, 231, 238
- \fntcolwidth 3, 117, 123
- \fnthours 2, 282
- \font 20, 25
- \fontdimen .. 27, 29–31
- \fontencoding 16
- \fontfamily 16
- \fontname 20
- \fontrange 3, 61
- \fontseries 16
- \fontshape 16
- \fonttable 2, 8
- \fonttext 3, 168
- \fonttexts . 3, 265, 265
- \frqq . 218, 230, 231, 238
- \ftable 12, 22, 127
- \ftablewidth 3, 117, 128
- \fulltext 3, 261
- G**
- \germanparatext . 3, 223
- \germantext 223
- \GLYPHALPHABET .. 4, 310
- \glyphalphabet .. 4, 310
- \glyphalternation 4, 298
- \glyphdigits 5, 317
- \glyphlowers 5, 317
- \glyphmixture ... 4, 291
- \glyphpunct 5, 330
- \glyphseries 4, 304
- \glyphuppers 317
- \guillemotleft 218
- \guillemotright ... 218
- H**
- \H 39, 214
- \h 40, 43, 55
- \hexoct 3, 47
- \hexoctfalse 49
- \hexocttrue 48
- \hline 99, 101, 109
- \hrulefill 54
- \ht 35, 151, 159
- I**
- \i 205, 214
- \iff@tskipping .. 59, 90
- \ifhexoct 47, 140
- \itshape 37
- J**
- \j 215
- K**
- \knunames 201, 261
- \knutext .. 172, 178, 181
- L**
- \L 214
- \l 214
- \lastpenalty 45

- | | | |
|--|---|---|
| <code>\latinparatext</code> .. 3, <u>242</u> | O | <code>\sixt@@n</code> 67, |
| <code>\latintext</code> <u>242</u> | <code>\O</code> 214 | 68, 72, 74, 87, 90 |
| <code>\loop</code> 34, 86, | <code>\o</code> 216 | <code>\smash</code> 55 |
| 295, 307, 321, 324 | <code>\OE</code> 205 | <code>\spaceskip</code> 29 |
| <code>\lower</code> 55, 117, 131 | <code>\oe</code> 205 | <code>\ss</code> 204, 237 |
| M | P | T |
| <code>\m</code> 282 | <code>\PackageError</code> 78 | <code>\tabskip</code> 129 |
| <code>\meaning</code> 88 | <code>\PackageWarning</code> 224, 243 | <code>\time</code> 283, 284 |
| <code>\MessageBreak</code> . 225, 244 | <code>\pikfont</code> 2, <u>15</u> , 19 | <code>\tiny</code> 158 |
| <code>\moreknutext</code> .. 197, <u>261</u> | <code>\providecommand</code> . 5, | <code>\ttfamily</code> 39 |
| <code>\multicolumn</code> 123 | 220, 221, 223, 242 | |
| <code>\multiply</code> 42, 68, 74, 284 | <code>\ProvidesPackage</code> ... 3 | U |
| <code>\multispan</code> 54 | <code>\punctext</code> 4, <u>277</u> | <code>\u</code> 214 |
| N | R | V |
| <code>\n</code> 282 | <code>\raise</code> 143 | <code>\v</code> 215 |
| <code>\newcount</code> 6 | <code>\regulartext</code> 3, <u>175</u> | <code>\vbox</code> 117, 165 |
| <code>\newif</code> 47, 59 | <code>\regulartexts</code> ... 3, <u>272</u> | <code>\vrule</code> 130 |
| <code>\next</code> 87, 88, 94, 95 | <code>\repeat</code> 34, 90, | |
| <code>\noalign</code> 53, 57, 105, 142 | 296, 308, 322, 325 | X |
| <code>\nodecimals</code> 3, <u>147</u> | S | <code>\xdef</code> 41, 42, 106 |
| <code>\nohexoct</code> 3, <u>47</u> | <code>\sevenrm</code> 5 | <code>\xfonttable</code> 2, <u>18</u> |
| <code>\nointerlineskip</code> 53, 57 | <code>\simpletext</code> 3, <u>261</u> | <code>\xspaceskip</code> 30 |
| <code>\normalfont</code> 20 | | |