

The fonttable package*

Peter Wilson[†]
Herries Press

2009/05/06

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
3	The code	3
	Bibliography	11

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.

`\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

*This file has version number v1.4, last revised 2009/05/06.

[†]herries dot press at earthlink dot net

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

Dingbats).

`\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, 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`

`\nodecimals`

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`

`\nohexoct`

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.

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

<code>\fetablewidth</code>	The font table's width is the length <code>\fetablewidth</code> , which by default is set to the normal textwidth (or more exactly, to <code>\hsize</code>). The table itself is left aligned. However, if <code>\nohexoct</code> is in effect the width of the table is its natural width.
<code>\fntcolwidth</code>	When <code>\nohexoct</code> is in effect the minimum width of a table column is <code>\fntcolwidth</code> . This is initially declared as <code>\setwidth{\fntcolwidth}{0.08\fetablewidth}</code>
<code>\fonttext</code>	The command <code>\fonttext{<testfont>}</code> typesets an example text using the <code><testfont></code> (e.g. <code>cmr10</code>).
<code>\simpletext</code>	The example text can be just a paragraph and a line of capitals, or include
<code>\fulltext</code>	more complex accented words as well. Following the declaration <code>\fulltext</code> the complex words are included as well as the example paragraph. The default is <code>\simpletext</code> for just the paragraph.
<code>\regulartext</code>	The command <code>\regulartext{<fontspec>}</code> typesets the example text using <code><fontspec></code> , for example <code>\rmfamily\itshape</code> or <code>\pikfont{T1}{pnc}{m}{it}</code> .
<code>\fonttexts</code>	The macro <code>\fonttexts{<testfont>}{<text>}</code> typesets <code><text></code> using the <code><testfont></code>
<code>\regulartexts</code>	(e.g. <code>cmr10</code>). Similarly the macro <code>\regulartexts{<fontspec>}{<text>}</code> typesets <code><text></code> using <code><fontspec></code> (e.g., <code>\rmfamily\itshape</code> or <code>\pikfont{T1}{ppl}{m}{it}</code>).
<code>\germantext</code>	<code>\germantext</code> expands to German language paragraph, borrowed from the
<code>\latintext</code>	<code>blindtext</code> package [Lik05]. <code>\latintext</code> expands to one version of a paragraph of the traditional <i>lorem ipsum</i> dummy Latin text. Either, or both, of these could be used as the <code><text></code> argument to <code>\fonttexts</code> or <code>\regulartexts</code> .
<code>\aztext</code>	<code>\aztext</code> expands to the lowercase Latin alphabet a to z, and <code>\AZtext</code> is the
<code>\AZtext</code>	corresponding command for the uppercase A to Z. The macros <code>\digitstext</code> and
<code>\digitstext</code>	<code>\puncttext</code> expand respectively to the digits 0 to 9, and to the typical punctuation
<code>\puncttext</code>	marks. In all cases there is a space between each character.

3 The code

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.

```

1 <*pack>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{fonttable}[2009/05/06 v1.4 displays a font]
4
\sevenrm A small fixed size roman font.
5 \providecommand*\sevenrm{\fontsize{7}{9pt}\rmfamily}

\f@tm Counts and a dimen.
\f@tn 6 \newcount\f@tm \newcount\f@tn \newcount\f@tp \newdimen\f@tdim
\f@tp 7
\f@tdim
\fonttable \fonttable{<font>} typesets a table of all the glyphs in the <font> (e.g., aunc10).
8 \newcommand*\fonttable[1]{%
9   \def\f@tfontname{#1}%
10  \bgroup
```

```

11 \f@tstartfont
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\f@tfontname{\fontname\font}\normalfont
21 \f@tstartfont
22 \ftable
23 \egroup}
24

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

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

\f@toct \f@toct{<onum>} typesets the octal constant <onum>.
37 \newcommand*{\f@toct}[1]{\hbox{\rmfamily\'}\kern-.2em\itshape
38 #1\kern.05em}} % octal constant

\f@thex \f@thex{<hnum>} typesets the hexadecimal constant <hnum>.
39 \newcommand*{\f@thex}[1]{\hbox{\rmfamily\H}\ttfamily#1}} % hexadecimal constant

\f@tsetdigs \f@tsetdigs
40 \def\f@tsetdigs#1"#2{\gdef\h{#2}% \h=hex prefix; \0\1=corresponding octal
41 \f@tm=\f@tn \divide\f@tm by 64 \xdef\0{\the\f@tm}%
42 \multiply\f@tm by-64 \advance\f@tm by\f@tn \divide\f@tm by 8 \xdef\1{\the\f@tm}}
```

```

\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

\fontoddlinenum \fontoddlne
52 \newcommand*{\fontoddlne}{\cr
53 \noalign{\nointerlineskip}
54 \multispan{19}\hrulefill&
55 \setbox0=\hbox{\lower 2.3pt\hbox{\fontthe{x}{h x}}}\smash{\box0}
56 \cr
57 \noalign{\nointerlineskip}}
58

\iff@tskipping
\fontskippingtrue 59 \newif\iff@tskipping
\fontskippingfalse 60

\fontrange \fontrange{<low>}{<high>} sets the character range to be output.
61 \newcommand*{\fontrange}[2]{%
62 \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.

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

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

71 \fontm=#2
72 \divide \fontm \sixt@@n
73 \advance \fontm \@ne
74 \multiply \fontm \sixt@@n
75 \ifnum \fontm > \ccclvi \fontm=\ccclvi \fi
76 \edef\fonthigh{\the\fontm}
77 \else

```

```

78 \PackageError{fonttable}{%
79     Improper values for fontrange. Default values substituted}{\@ehc}
80 \def\f@tlow{0} \def\f@thigh{256}
81 \fi}
82 \fontrange{0}{256}
83

\@tloopforsixteen \f@tloopforsixteen sets up a block of sixteen character slots.
84 \newcommand*\f@tloopforsixteen{%
85     \ifnum\f@tn<\f@tlow \global\f@tn=\f@tlow\fi
86     \loop\f@tskippingfalse
87     \ifnum\f@tn<\f@thigh \f@tm=\f@tn \divide\f@tm \sixt@@n \chardef\next=\f@tm
88     \expandafter\f@tsetdigs\meaning\next \f@ttestrow
89     \ifnum\f@tp=\@one \f@tskippingtrue \fi\fi
90     \iff@tskipping \global\advance\f@tn \sixt@@n \repeat}
91

\@tevenline \f@tevenline gets next non-empty set of a block of 16 characters. It either calls
\f@tevenlinenonum \f@tmorechart to print them, or \f@tendchart to finish off the table if all 256
potential characters have been processed.
    \f@tevenlinenonum does something similar when no external numbers are
printed.
92 \newcommand*\f@tevenline{%
93     \f@tloopforsixteen
94     \ifnum\f@tn=\f@thigh \let\next=\f@tendchart\else\let\next=\f@tmorechart\fi
95     \next}
96 \newcommand*\f@tevenlinenonum{%
97     \f@tloopforsixteen
98     \ifnum\f@tn=\f@thigh
99         \\hline
100     \else
101         \\hline
102         \f@tmorechartnonum
103     \fi}
104

\@tmorechart \f@tmorechart sets two lines of the table, and \f@tmorechartnonum does the
\f@tmorechartnonum same when there are no external numbers.
105 \newcommand*\f@tmorechart{\cr\noalign{\hrule\penalty5000}
106 \f@tchartline \f@toddlne \f@tm=\@one \advance\f@tm \@one \xdef\@the\f@tm{
107 \f@tchartline \f@tevenline}
108 \newcommand*\f@tmorechartnonum{%
109 \f@tsimpleline \\ \hline
110 \f@tsimpleline \f@tevenlinenonum}
111

\@tchartline \f@tchartline does a line of the table, including external numbers, and
\f@tsimpleline \f@tsimpleline does an unnumbered line.
112 \newcommand*\f@tchartline{%

```

```

113 &\f@toct{\0\1x}&&\f@tpsg{}&&\f@tpsg{}&&\f@tpsg{}&&\f@tpsg{}&&\f@tpsg{}&&\f@tpsg{}&&\f@tp
114 \newcommand*\f@tsimpleline}{%
115 \f@tpsg{}\f@tchartstrut& \f@tpsg{} & \f@tpsg{} & \f@tpsg{} & \f@tpsg{} & \f@tpsg{} & \f@tp
116

```

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

```

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

```

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

```

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

```

`\f@table` `\f@table` 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\f@tablewidth\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*\f@table}{\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 `\nodecimals` 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*\nodedecimal{%
148 %%% the original
149 \newcommand*\f@tpsg{\setbox0=\hbox{\char\f@tn}%
150 \ifdim\ht0>7.5pt\f@treposition
151 \else\ifdim\dp0>2.5pt\f@treposition\fi\fi
152 \box0\global\advance\f@tn 1 }}
153 \newcommand*\decimals{%
154 %%% mine
155 \newcommand*\f@tpsg{\setbox0=\hbox{\char\f@tn,\rlap{\tiny \the\f@tn}}%
156 \ifdim\ht0>7.5pt\f@treposition
157 \else\ifdim\dp0>2.5pt\f@treposition\fi\fi
158 \box0\global\advance\f@tn 1 }}
159 \decimals
160

\f@treposition \f@treposition

161 \newcommand*\f@treposition{\setbox0=\vbox{\kern2pt\box0}\f@tdim=\dp0
162 \advance\f@tdim 2pt \dp0=\f@tdim}
163

\fonttext \fonttext{<font>} typesets \knutext using <font> (e.g. aunc10).

164 \def\fonttext#1{%
165 \def\f@tfontname{#1}%
166 \bgroup
167 \f@tstartfont
168 \knutext
169 \egroup}
170

\regulartext \regulartext{<fontspec>} typesets \knutext using <fontspec> (e.g., \aunc1family).

171 \def\regulartext#1{%
172 \bgroup
173 #1
174 \knutext
175 \egroup}
176

\knutext Deathless prose from Knuth for testing a font. It includes \moreknutext,
\capknutext, and \knunames.

177 \def\knutext{
178 On November 14, 1885, Senator \& Mrs.~Leland Stanford called together
179 at their San Francisco mansion the 24~prominent men who had been
180 chosen as the first trustees of The Leland Stanford Junior University.
181 They handed to the board the Founding Grant of the University, which
182 they had executed three days before. This document---with various

```


183 amendments, legislative acts, and court decrees---remains as the
 184 University's charter. In bold, sweeping language it stipulates that
 185 the objectives of the University are 'to qualify students for
 186 personal success and direct usefulness in life; and to promote the
 187 publick welfare by exercising an influence in behalf of humanity and
 188 civilization, teaching the blessings of liberty regulated by law, and
 189 inculcating love and reverence for the great principles of government
 190 as derived from the inalienable rights of man to life, liberty, and
 191 the pursuit of happiness.'

192
 193 \moreknutext
 194
 195 \capknutext
 196
 197 \knunames
 198 \par}}
 199

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

200 \def\@moreknutext{?'But aren't Kafka's Schlo{\ss} and {\AE}sop's
 201 {\OE}uvres often na{"i}ve vis-\`a-vis the d{\ae}monic ph{\oe}nix's
 202 official r\`ole in fluffy souffl\`es? }
 203

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

\capknutext 204 \newcommand{\@capknutext}{%
 205 (!'THE DAZED BROWN FOX QUICKLY GAVE 12345--67890 JUMPS!)}
 206 \let\capknutext\@capknutext
 207

\@knunames Lots of accents masquerading in personal names.

208 \def\@knunames{ {\AA}ngel\aa\ Beatrice Claire
 209 Diana \`Erica Fran\c{c}oise Ginette H\`el\`ene Iris
 210 Jackie K\=aren {\L}au\ra Mar{\`i}a N\H{a}ta{\l}{\u{i}}e {\O}ctave
 211 Pauline Qu\`eneau Roxanne Sabine T\~a{\`j}a Ur\v{s}ula
 212 Vivian Wendy Xanthippe Yv{\o}nne Z\"azilie\par}
 213

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

\flqq 214 \DeclareTextSymbol{\guillemotleft}{OT1}{'\'}
 \frqq 215 \DeclareTextSymbol{\guillemotright}{OT1}{'\'}
 216 \providecommand{\flqq}{\guillemotleft}
 217 \providecommand{\frqq}{\guillemotright}
 218

\germantext Text from the Blindtext package.

219 \newcommand*\germantext{%
 220 Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer

```

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

```

`\latintext` The traditional printers' text.

```

235 \newcommand*{\latintext}{%
236 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam
237 lobortis facilisis sem. Nullam nec mi et neque pharetra
238 sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper,
239 felis non sodales commodo, lectus velit ultrices augue, a
240 dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie
241 ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in
242 sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
243 Duis fringilla tristique neque. Sed interdum libero ut metus.
244 Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit
245 amet ante lobortis sollicitudin. Praesent blandit blandit mauris.
246 Praesent lectus tellus, aliquet aliquam, luctus a, egestas a,
247 turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum
248 turpis accumsan semper.\par}
249

```

`\simpletext` `\simpletext` kills off `\morekntext` and `\knunames`. `\fulltext` restores `\morekntext` and `\knunames`. Make `\fulltext` the default.

```

\morekntext 250 \newcommand*{\simpletext}{\let\morekntext\relax \let\knunames\relax}
\knunames 251 \newcommand*{\fulltext}{\let\morekntext\@morekntext \let\knunames\@knunames}
252 \fulltext
253

```

`fonttexts` `\fonttexts{}{<text>}` typesets `<text>` using `` (e.g. `auncl10`).

```

254 \def\fonttexts#1#2{%
255   \def\fontname{#1}%
256   \bgroup
257   \fontstartfont
258   #2
259   \egroup}
260

```

`\regulartexts` `\regulartext{<fontspec>}{<text>}` typesets `<text>` using `<fontspec>` (e.g., `\aunclfamily`).

```

261 \def\regulartexts#1#2{%
262   \bgroup
263   #1 #2
264   \egroup}
265
\aztext  The various characters used for Latin texts.
\AZtext 266 \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 267 \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 268 \newcommand*{\digitstext}{0 1 2 3 4 5 6 7 8 9}
269 \newcommand*{\puncttext}{' ! @ \$ \% & * ( ) \_ - + = [ ] < > \{ \} : ; ' , . ? /}
270
The end of the package.
271 \</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.

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				
\"	201, 212, 223, 227, 228	_ 269	\AZtext 3, <u>266</u>
\\$ 269	\‘ 201, 209, 215	\aztext 3, <u>266</u>
\& 178, 269	\~ 211		
\, 155			B	
\- 223, 227, 228	Numbers		\baselineskip 35
\/ 38	\0 40, 41, 113	C	
\= 210	\1 40, 42, 106, 113	\c 209
\@capknuttext <u>204</u>			\capknuttext	... 195, <u>204</u>
\@cclvi 75	_	... 208, 224, 230, 231	\char	... 34, 43, 149, 155
\@ehc 79			\chardef 87
\@knunames	... <u>208</u> , 251	A			
\@moreknuttext	... <u>200</u> , 251	\AA 208	D	
\{ 269	\aa 208	\decimals 2, <u>147</u>
\} 269	\AE 200	\DeclareRobustCommand	
\^ 202, 211	\ae 201	15

- `\DeclareTextSymbol` 214, 215
- `\digitstext` 3, 266
- `\dp` 35, 151, 157, 161, 162
- E**
- `\edef` 20, 70, 76
- F**
- `\f@tchartline` 106, 107, 112
- `\f@tchartstrut` 115, 117, 129
- `\f@tcol` 122
- `\f@tdim` 6, 161, 162
- `\f@tendchart` 94, 142
- `\f@tevenline` 92, 107, 133
- `\f@tevenlinenonum` 92, 110, 138
- `\f@tfontname` 9, 20, 25, 165, 255
- `\f@tftablenonum` 127
- `\f@tftablenum` 127
- `\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@tmorerechart` 94, 105
- `\f@tmorerechartnonum` 102, 105
- `\f@tn` . . 6, 33, 34, 41, 42, 85, 87, 90, 94, 98, 127, 135, 149, 152, 155, 158
- `\f@toct` . . . 37, 113, 132
- `\f@todddline` 52, 106
- `\f@todddlinenum` 52
- `\f@tp` 6, 45, 89
- `\f@tpsg` . . . 113, 115, 147
- `\f@treposition` 150, 151, 156, 157, 161
- `\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, 167, 257
- `\f@ttestfont` 25–27, 29–31
- `\f@ttestrow` 43, 88
- `\flqq` 214, 224, 231
- `\fntcolwidth` 3, 117, 123
- `\font` 20, 25
- `\fontdimen` . . 27, 29–31
- `\fontencoding` 16
- `\fontfamily` 16
- `\fontname` 20
- `\fonrange` 2, 61
- `\fontseries` 16
- `\fontshape` 16
- `\fonttable` 1, 8
- `\fonttext` 3, 164
- `\fonttexts` . . 3, 254, 254
- `\frqq` . 214, 223, 224, 231
- `\ftable` 12, 22, 127
- `\ftablewidth` 3, 117, 128
- `\fulltext` 3, 250
- G**
- `\germantext` 3, 219
- `\guillemotleft` 214
- `\guillemotright` 214
- H**
- `\H` 39, 210
- `\h` 40, 43, 55
- `\hexoct` 2, 47
- `\hexoctfalse` 49
- `\hexocttrue` 48
- `\hline` 99, 101, 109
- `\hrulefill` 54
- `\ht` 35, 150, 156
- I**
- `\i` 201, 210
- `\iff@tskipping` . . 59, 90
- `\ifhexoct` 47, 140
- `\itshape` 37
- J**
- `\j` 211
- K**
- `\knunames` 197, 250
- `\ktext` . . 168, 174, 177
- L**
- `\L` 210
- `\l` 210
- `\lastpenalty` 45
- `\latintext` 3, 235
- `\loop` 34, 86
- `\lower` 55, 117, 131
- M**
- `\meaning` 88
- `\morektext` . . 193, 250
- `\multicolumn` 123
- `\multiply` . . . 42, 68, 74
- `\multispan` 54
- N**
- `\newcount` 6
- `\newif` 47, 59
- `\next` 87, 88, 94, 95
- `\noalign` 53, 57, 105, 142
- `\nodecimals` 2, 147
- `\nohexoct` 2, 47
- `\nointerlineskip` 53, 57
- `\normalfont` 20
- O**
- `\O` 210
- `\o` 212
- `\OE` 201
- `\oe` 201
- P**
- `\PackageError` 78
- `\pikfont` 2, 15, 19
- `\providecommand` 5, 216, 217
- `\ProvidesPackage` 3
- `\puncttext` 3, 266
- R**
- `\raise` 143
- `\regulartext` 3, 171
- `\regulartexts` 3, 261
- `\repeat` 34, 90
- S**
- `\sevenrm` 5

<code>\simpletext</code> 3, <u>250</u>	<code>\tiny</code> 155	<code>\vrule</code> 130
<code>\sixt@cn</code> 67, 68, 72, 74, 87, 90	<code>\ttfamily</code> 39	
<code>\smash</code> 55	U	X
<code>\spaceskip</code> 29	<code>\u</code> 210	
<code>\ss</code> 200, 230	V	<code>\xdef</code> 41, 42, 106
T	<code>\v</code> 211	<code>\xfonttable</code> 2, <u>18</u>
<code>\tabskip</code> 129	<code>\vbox</code> 117, 161	<code>\xspaceskip</code> 30