Name

groff_font – GNU *roff* device and font description files

Description

The *groff* font and output device description formats are slight extensions of those used by AT&T device-independent *troff*. In distinction to the AT&T implementation, *groff* lacks a binary format; all files are text files. (Plan 9 *troff* has also abandoned the binary format.) The device and font description files for a device *name* are stored in a *dev*name directory. The device description file is called *DESC*, and, for each font supported by the device, a font description file is called *f,* where *f* is usually an abbreviation of a font's name and/or style. For example, the **ps** (PostScript) device has *groff* font description files for Times roman (**TR**) and Zapf Chancery Medium italic (**ZCMI**), among many others, while the **utf8** device (for terminal emulators) has only font descriptions for the roman, italic, bold, and bold-italic styles (**R**, **I**, **B**, and **BI**, respectively).

Device and font description files are read by the formatter, *troff*, and by output drivers. The programs typically delegate these files' processing to an internal library, *libgroff*, ensuring their consistent interpretation.

DESC file format

The *DESC* file contains a series of directives; each begins a line. Their order is not important, with two exceptions: (1) the **res** directive must precede any **papersize** directive; and (2) the **charset** directive must come last (if at all). If a directive name is repeated, later entries in the file override previous ones (except that the paper dimensions are computed based on the **res** directive last seen when **papersize** is encountered). Spaces and/or tabs separate words and are ignored at line boundaries. Comments start with the "**#**" character and extend to the end of a line. Empty lines are ignored.

family *fam*
The default font family is *fam*.

fonts *n F1 … Fn*
Fonts *F1*, …, *Fn* are mounted at font positions $m + 1$, …, $m + n$ where *m* is the number of **styles** (see below). This directive may extend over more than one line. A font name of **0** causes no font to be mounted at the corresponding position.

hor *n* The horizontal motion quantum is *n* basic units. Horizontal quantities are rounded to multiples of *n*.

image_generator *program*
Use *program* to generate PNG images from PostScript input. Under GNU/Linux, this is usually *gs*(1), but under other systems (notably Cygwin) it might be set to another name. The *grohtml*(1) driver uses this directive.

paperlength *n*
The vertical dimension of the output medium is *n* basic units (deprecated: use **papersize** instead).

papersize *format-or-dimension-pair-or-file-name …*
The dimensions of the output medium are as according to the argument, which is either a standard paper format, a pair of dimensions, or the name of a plain text file containing either of the foregoing. Recognized paper formats are the ISO and DIN formats **A0–A7**, **B0–B7**, **C0–C7**, and **D0–D7**; the U.S. formats **letter**, **legal**, **tabloid**, **ledger**, **statement**, and **executive**; and the envelope formats **com10**, **monarch**, and **DL**. Matching is performed without regard for lettercase.

Alternatively, the argument can be a custom paper format *length***,***width* (with no spaces before or after the comma). Both *length* and *width* must have a unit appended; valid units are "**i**" for inches, "**c**" for centimeters, "**p**" for points, and "**P**" for picas. Example: "**12c,235p**". An argument that starts with a digit is always treated as a custom paper format.

Finally, the argument can be a file name (e.g., */etc/papersize*); if the file can be opened, the first line is read and a match attempted against each other form. No comment syntax is supported.

More than one argument can be specified; each is scanned in turn and the first valid paper specification used.

paperwidth *n*
: The horizontal dimension of the output medium is *n* basic units (deprecated: use **papersize** instead).

pass_filenames
: Direct *troff* to emit the name of the source file being processed. This is achieved with the intermediate output command "**x F**", which *grohtml* interprets.

postpro *program*
: Use *program* as the postprocessor.

prepro *program*
: Use *program* as a preprocessor. The **html** and **xhtml** output devices use this directive.

print *program*
: Use *program* as the print spooler. If omitted, *groff*'s **–l** and **–L** options are ignored.

res *n*
: The device resolution is *n* basic units per inch.

sizes *s1* ... *sn* **0**
: The device has fonts at *s1*, ..., *sn* scaled points (see below). The list of sizes must be terminated by a **0**. Each *si* can also be a range of sizes *m*–*n*. The list can extend over more than one line.

sizescale *n*
: A typographical point is subdivided into *n* scaled points. The default is **1**.

styles *S1* ... *Sm*
: The first *m* font mounting positions are associated with styles *S1*, ..., *Sm*.

tcommand
: The postprocessor can handle the **t** and **u** intermediate output commands.

unicode
: The output device supports the complete Unicode repertoire. This directive is useful only for devices which produce character entities instead of glyphs.

  If **unicode** is present, no **charset** section is required in the font description files since the Unicode handling built into *groff* is used. However, if there are entries in a font description file's **charset** section, they either override the default mappings for those particular characters or add new mappings (normally for composite characters).

  The **utf8**, **html**, and **xhtml** output devices use this directive.

unitwidth *n*
: Quantities in the font description files are in basic units for fonts whose type size is *n* scaled points.

unscaled_charwidths
: Make the font handling module always return unscaled glyph widths. The *grohtml* driver uses this directive.

use_charnames_in_special
: *troff* should encode named glyphs inside device control commands. The *grohtml* driver uses this directive.

vert *n*
: The vertical motion quantum is *n* basic units. Vertical quantities are rounded to multiples of *n*.

charset
: This directive and the rest of the file are ignored. It is recognized for compatibility with other *troff* implementations. In GNU *troff*, character set repertoire is described on a per-font basis.

troff recognizes but ignores the directives **spare1**, **spare2**, and **biggestfont**.

The **res**, **unitwidth**, **fonts**, and **sizes** lines are mandatory. Directives not listed above are ignored by *troff* but may be used by postprocessors to obtain further information about the device.

Font description file format

On typesetting output devices, each font is typically available at multiple sizes. While paper measurements in the device description file are in absolute units, measurements applicable to fonts must be proportional to the type size. *groff* achieves this using the precedent set by AT&T device-independent *troff*: one font size is chosen as a norm, and all others are scaled linearly relative to that basis. The "unit width" is the number of basic units per point when the font is rendered at this nominal size.

For instance, *groff*'s **lbp** device uses a **unitwidth** of 800. Its Times roman font ("**TR**") has a **spacewidth** of 833; this is also the width of its comma, period, centered period, and mathematical asterisk, while its "M" is 2,963 basic units. Thus, an "M" on the **lbp** device is 2,963 basic units wide at a notional type size of 800 points. (800-point type is not practical for most purposes, but using it enables the quantities in the font description files to be expressed as integers.)

A font description file has two sections. The first is a sequence of directives, and is parsed similarly to the *DESC* file described above. Except for the directive names that begin the second section, their ordering is immaterial. Later directives of the same name override earlier ones, spaces and tabs are handled in the same way, and the same comment syntax is supported. Empty lines are ignored throughout.

name *F*
: The name of the font is *F*. "**DESC**" is an invalid font name. Simple integers are valid, but their use is discouraged. (*groff* requests and escape sequences interpret non-negative font names as mounting positions instead. Further, a font named "**0**" cannot be automatically mounted by the **fonts** directive of a *DESC* file.)

spacewidth *n*
: The width of an unadjusted inter-word space is *n* basic units.

The directives above must appear in the first section; those below are optional.

slant *n* The font's glyphs have a slant of *n* degrees; a positive *n* slants in the direction of text flow.

ligatures *lig1* ... *lign* [**0**]
: Glyphs *lig1*, ..., *lign* are ligatures; possible ligatures are **ff**, **fi**, **fl**, **ffi**, and **ffl**. For compatibility with other *troff* implementations, the list of ligatures may be terminated with a **0**. The list of ligatures must not extend over more than one line.

special The font is *special*: when a glyph is requested that is not present in the current font, it is sought in any mounted fonts that bear this property.

Other directives in this section are ignored by *troff*, but may be used by postprocessors to obtain further information about the font.

The second section contains one or two subsections. These can appear in either order; the first one encountered commences the second section. Each starts with a directive on a line by itself. A **charset** subsection is mandatory unless the associated *DESC* file contains the **unicode** directive. Another subsection, **kernpairs**, is optional.

The directive **charset** starts the character set subsection. (For typesetter devices, this directive is misnamed since it starts a list of glyphs, not characters.) It precedes a series of glyph descriptions, one per line. Each such glyph description comprises a set of fields separated by spaces or tabs and organized as follows.

name metrics type code [*entity-name*] [**--** *comment*]

name identifies the glyph: if *name* is a printable character *c*, it corresponds to the *troff* ordinary character *c*. If *name* is a multi-character sequence not beginning with ****, it corresponds to the GNU *troff* special character escape sequence "**\[***name***]**". A name consisting of three minus signs, "**---**", indicates that the glyph is unnamed: such glyphs can be accessed only by the **\N** escape sequence in *troff*. A special character named "**---**" can still be defined using **.char** and similar requests. The *name* "**\-**" defines the minus sign glyph. Finally, *name* can be the horizontal motion escape sequences, **\|** and **\^** ("thin" and "hair" spaces, respectively), in which case only the width metric described below is applied; a font can thus customize the widths of these spaces.

The form of the *metrics* field is as follows (on one line; it may be broken here for readability).

> *width*[**,**[*height*[**,**[*depth*[**,**[*italic-correction*[**,**[*left-italic-correction*[**,**[*subscript-correction*]]]]]]]]]]

There must not be any spaces, tabs, or newlines between these *subfields,* which are in basic units expressed as decimal integers. Unspecified subfields default to **0**. Since there is no associated binary format, these values are not required to fit into the C language data type **char** as they are in AT&T device-independent *troff*.

The *width* subfield gives the width of the glyph. The *height* subfield gives the height of the glyph (upwards is positive); if a glyph does not extend above the baseline, it should be given a zero height, rather than a negative height. The *depth* subfield gives the depth of the glyph, that is, the distance below the baseline to which the glyph extends (downwards is positive); if a glyph does not extend below the baseline, it should be given a zero depth, rather than a negative depth. Italic corrections are relevant to glyphs in italic or oblique styles. The *italic-correction* is the amount of space that should be added after an oblique glyph to be followed immediately by an upright glyph. The *left-italic-correction* is the amount of space that should be added before an oblique glyph to be preceded immediately by an upright glyph. The *subscript-correction* is the amount of space that should be added after an oblique glyph to be followed by a subscript; it should be less than the italic correction.

For fonts used with typesetting devices, the *type* field gives a featural description of the glyph: it is a bit mask recording whether the glyph is an ascender, descender, both, or neither. When a **\w** escape sequence is interpolated, these values are bitwise or-ed together for each glyph and stored in the **ct** register. In font descriptions for terminal devices, all glyphs might have a type of zero, regardless of their appearance.

0 means the glyph lies entirely between the baseline and a horizontal line at the "x-height" of the font, as with "a", "c", and "x";

1 means the glyph descends below the baseline, like "p";

2 means the glyph ascends above the font's x-height, like "A" or "b"); and

3 means the glyph is both an ascender and a descender—this is true of parentheses in some fonts.

The *code* field gives a numeric identifier that the postprocessor uses to render the glyph. The glyph can be specified to *troff* using this code by means of the **\N** escape sequence. The code can be any integer (that is, any integer parsable by the C standard library's *strtol*(3) function).

The *entity-name* field defines an identifier for the glyph that the postprocessor uses to print the *troff* glyph *name*. This field is optional; it was introduced so that the *grohtml* output driver could encode its character set. For example, the glyph **\[Po]** is represented by "**&pound;**" in HTML 4.0. For efficiency, these data are now compiled directly into *grohtml*. *grops* uses the field to build sub-encoding arrays for PostScript fonts containing more than 256 glyphs. Anything on the line after the *entity-name* field or "**--**" is ignored.

A line in the **charset** section can also have the form

> *name* **"**

identifying *name* as another name for the glyph mentioned in the preceding line. Such aliases can be chained.

The directive **kernpairs** starts a list of kerning adjustments to be made to adjacent glyph pairs from this font. It contains a sequence of lines formatted as follows.

> *g1 g2 n*

The foregoing means that when glyph *g1* is typeset immediately before *g2*, the space between them should be increased by *n*. Most kerning pairs should have a negative value for *n*.

Files

*/usr/local/share/groff/1.23.0/font/dev*name*/DESC*
describes the output device *name*.

*/usr/local/share/groff/1.23.0/font/dev*name/F
describes the font known as *F* on device *name*.

See also

Groff: The GNU Implementation of troff, by Trent A. Fisher and Werner Lemberg, is the primary *groff* manual. You can browse it interactively with "info groff".

"Troff User's Manual" by Joseph F. Ossanna, 1976 (revised by Brian W. Kernighan, 1992), AT&T Bell Laboratories Computing Science Technical Report No. 54, widely called simply "CSTR #54", documents the language, device and font description file formats, and device-independent output format referred to collectively in *groff* documentation as "AT&T *troff*".

"A Typesetter-independent TROFF" by Brian W. Kernighan, 1982, AT&T Bell Laboratories Computing Science Technical Report No. 97, provides additional insights into the device and font description file formats and device-independent output format.

groff(1), subsection "Utilities", lists programs available for describing fonts in a variety of formats such that *groff* output drivers can use them.

troff(1) documents the default device and font description file search path.

groff_out(5), *addftinfo*(1)