

NAME

groff_font – format of groff device and font description files

DESCRIPTION

The groff font format is roughly a superset of the ditroff font format. The font files for device *name* are stored in a directory *devname*. There are two types of file: a device description file called *DESC* and for each font *F*, a font file called *F*. These are text files; unlike the ditroff font format, there is no associated binary format.

DESC file format

The *DESC* file can contain the following types of line as shown below. Later entries in the file override previous values.

Empty lines are ignored.

charset

This line and everything following in the file are ignored. It is allowed for the sake of backwards compatibility.

family *fam*

The default font family is *fam*.

fonts *n* *F1* *F2* *F3* ... *F_n*

Fonts *F1*, ..., *F_n* are mounted in the font positions *m* + 1, ..., *m* + *n* where *m* is the number of styles. This command may extend over more than one line. A font name of **0** causes no font to be mounted on the corresponding font position.

hor *n* The horizontal resolution is *n* machine units.**image_generator** *string*

Needed for **grohtml** only. It specifies the program to generate PNG images from PostScript input. Under GNU/Linux this is usually *gs* but under other systems (notably cygwin) it might be set to another name.

paperlength *n*

The physical vertical dimension of the output medium in machine units. This isn't used by **troff** itself but by output devices. Deprecated. Use **papersize** instead.

papersize *string*

Select a paper size. Valid values for *string* are the ISO paper types A0–A7, B0–B7, C0–C7, D0–D7, DL, and the US paper types letter, legal, tabloid, ledger, statement, executive, com10, and monarch. Case is not significant for *string* if it holds predefined paper types. Alternatively, *string* can be a file name (e.g. */etc/papersize*); if the file can be opened, **groff** reads the first line and tests for the above paper sizes. Finally, *string* can be a custom paper size in the format *length,width* (no spaces before and after the comma). Both *length* and *width* must have a unit appended; valid values are 'i' for inches, 'c' for centimeters, 'p' for points, and 'P' for picas. Example: **12c,235p**. An argument which starts with a digit is always treated as a custom paper format. **papersize** sets both the vertical and horizontal dimension of the output medium.

More than one argument can be specified; **groff** scans from left to right and uses the first valid paper specification.

paperwidth *n*

The physical horizontal dimension of the output medium in machine units. Deprecated. Use **papersize** instead. This isn't used by **troff** itself but by output devices.

pass_filenames

Make troff tell the driver the source file name being processed. This is achieved by another tcommand: **F** *filename*.

postpro *program*

Use *program* as the postprocessor.

prepro *program*

Call *program* as a preprocessor.

print *program*

Use *program* as the spooler program for printing. If omitted, the **-I** and **-L** options of **groff** are ignored.

res *n* There are *n* machine units per inch.

sizes *s1 s2 ... sn 0*

This means that the device has fonts at *s1*, *s2*, ..., *sn* scaled points. 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*

The scale factor for point sizes. By default this has a value of 1. One *scaled point* is equal to one point / *n*. The arguments to the **unitwidth** and **sizes** commands are given in scaled points.

styles *S1 S2 ... Sm*

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

tcommand

This means that the postprocessor can handle the **t** and **u** output commands.

unicode

Indicate that the output device supports the complete Unicode repertoire. 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 **charset** section, they either override the default mappings for those particular characters or add new mappings (normally for composite characters).

This is used for **-Tutf8**, **-Thtml**, and **-Txhtml**.

unitwidth *n*

Quantities in the font files are given in machine units for fonts whose point size is *n* scaled points.

unscaled_charwidths

Make the font handling module always return unscaled glyph widths. Needed for the **grohtml** device.

use_charnames_in_special

This command indicates that troff should encode named glyphs inside special commands.

vert *n* The vertical resolution is *n* machine units.

The **res**, **unitwidth**, **fonts**, and **sizes** lines are compulsory. Not all commands in the *DESC* file are used by **troff** itself; some of the keywords (or even additional ones) are used by postprocessors to store arbitrary information about the device.

Here a list of obsolete keywords which are recognized by **groff** but completely ignored: **spare1**, **spare2**, **biggestfont**.

Font file format

A font file has two sections; empty lines are ignored in both of them.

The first section is a sequence of lines each containing a sequence of blank delimited words; the first word in the line is a key, and subsequent words give a value for that key.

ligatures *lig1 lig2 ... lign [0]*

Glyphs *lig1*, *lig2*, ..., *lign* are ligatures; possible ligatures are **ff**, **fi**, **fl**, **ffi**, and **ffl**. For backwards compatibility, the list of ligatures may be terminated with a **0**. The list of ligatures may not extend over more than one line.

name *F*

The name of the font is *F*.

slant *n* The glyphs of the font have a slant of *n* degrees. (Positive means forward.)

spacewidth *n*

The normal width of a space is *n*.

special The font is *special*; this means that when a glyph is requested that is not present in the current font, it is searched for in any special fonts that are mounted.

Other commands are ignored by **troff** but may be used by postprocessors to store arbitrary information about the font in the font file.

The first section can contain comments which start with the **#** character and extend to the end of a line.

The second section contains one or two subsections. It must contain a *charset* subsection and it may also contain a *kernpairs* subsection. These subsections can appear in any order. Each subsection starts with a word on a line by itself.

The word **charset** starts the charset subsection. The **charset** line is followed by a sequence of lines. Each line gives information for one glyph. A line comprises a number of fields separated by blanks or tabs. The format is

```
name metrics type code [entity_name] [--- comment]
```

name identifies the glyph: if *name* is a single glyph *c* then it corresponds to the groff input character *c*; if it is of the form *\c* where *c* is a single character, then it corresponds to the special character *\[c]*; otherwise it corresponds to the groff input character *\[name]*. If it is exactly two characters *xx* it can be entered as *\(xx*. Note that single-letter special characters can't be accessed as *\c*; the only exception is *\-* which is identical to *\[-]*. The name *---* is special and indicates that the glyph is unnamed; such glyphs can only be used by means of the *\N* escape sequence in **troff**.

The *type* field gives the glyph type:

- 1 means the glyph has a descender, for example, 'p';
- 2 means the glyph has an ascender, for example, 'b';
- 3 means the glyph has both an ascender and a descender, for example, 'c'.

The *code* field gives the code which the postprocessor uses to print the glyph. The glyph can also be input to groff using this code by means of the *\N* escape sequence. The code can be any integer. If it starts with a **0** it is interpreted as octal; if it starts with **0x** or **0X** it is interpreted as hexadecimal. Note, however, that the *\N* escape sequence only accepts a decimal integer.

The *entity_name* field gives an ASCII string identifying the glyph which the postprocessor uses to print that glyph. This field is optional and is currently used by **gr ops** to build sub-encoding arrays for PS fonts containing more than 256 glyphs. (It has also been used for **grohtml**'s entity names but for efficiency reasons this data is now compiled directly into **grohtml**.)

Anything on the line after the encoding field or *---* are ignored.

The *metrics* field has the form (in one line; it is broken here for the sake of readability):

```
width[,height[,depth[,italic-correction
[,left-italic-correction[,subscript-correction]]]]]
```

There must not be any spaces between these subfields. Missing subfields are assumed to be 0. The subfields are all decimal integers. Since there is no associated binary format, these values are not required to fit into a variable of type **char** as they are in ditroff. The *width* subfields give 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. The *italic-correction* subfield gives the amount of space that should be added after the glyph when it is

immediately to be followed by a glyph from a roman font. The *left-italic-correction* subfield gives the amount of space that should be added before the glyph when it is immediately to be preceded by a glyph from a roman font. The *subscript-correction* gives the amount of space that should be added after a glyph before adding a subscript. This should be less than the italic correction.

A line in the charset section can also have the format

name "

This indicates that *name* is just another name for the glyph mentioned in the preceding line.

The word **kernpairs** starts the kernpairs section. This contains a sequence of lines of the form:

c1 c2 n

This means that when glyph *c1* appears next to glyph *c2* the space between them should be increased by *n*. Most entries in kernpairs section have a negative value for *n*.

FILES

/usr/share/groff/1.22.4/font/devname/DESC

Device description file for device *name*.

/usr/share/groff/1.22.4/font/devname/F

Font file for font *F* of device *name*.

SEE ALSO

groff_out(5), **troff(1)**, **addftinfo(1)**, **afmtodit(1)**

A man page *name(n)* of section *n* can be viewed either with

\$ **man** *n name*

for text mode or

\$ **groffer** *n name*

for graphical mode (default is PDF mode).