## 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 **dev***name.* 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.

**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 . . . Fn*

Fonts *F1 . . . Fn* will be 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** will cause no font to be mounted on the corresponding font position.

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

**paperheight** *n*

The physical vertical dimension of the output medium in machine units. This isn't used by **troff** itself; currently, only **grops** uses it.

**paperwidth** *n*

The physical horizontal dimension of the output medium in machine units. This isn't used by **troff**. Currently, only the **grolbp** output device uses it.

**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.

**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 **−l** 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 pointsizes. 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 will be associated with styles *S1 . . . Sm*.

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

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

**use_charnames_in_special**
This command indicates that troff should encode named characters inside special commands.

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

The **res**, **unitwidth**, **fonts**, and **sizes** lines are compulsory. Other commands are ignored by **troff** but may be used by postprocessors to store arbitrary information about the device in the DESC file.

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. 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**]
Characters *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 characters 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 character is requested that is not present in the current font, it will be 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 character. 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 character: if *name* is a single character *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 character is unnamed; such characters can only be used by means of the **\N** escape sequence in **troff**.

Groff supports eight-bit characters; however some utilities have difficulties with eight-bit characters. For this reason, there is a convention that the name **char***n* is equivalent to the single character whose code is *n*. For example, **char163** would be equivalent to the character with code 163 which is the pounds sterling sign in ISO Latin-1.

The *type* field gives the character type:

1 means the character has a descender, for example, p;

2 means the character has an ascender, for example, b;

3 means the character has both an ascender and a descender, for example, (.

The *code* field gives the code which the postprocessor uses to print the character. The character 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 will be interpreted as octal; if it starts with **0x** or **0X** it will be intepreted 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 the character. This field is optional and has been introduced so that the html device driver can encode its character set. For example, the character '\[Po]' is represented as '£' in html 4.0.

Anything on the line after the encoding field resp. after '--' will be 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 gives the width of the character. The *height* subfield gives the height of the character (upwards is positive); if a character 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 character, that is, the distance below the lowest point below the baseline to which the character extends (downwards is positive); if a character does not extend below above 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 character when it is immediately to be followed by a character from a roman font. The *left-italic-correction* subfield gives the amount of space that should be added before the character when it is immediately to be preceded by a character from a roman font. The *subscript-correction* gives the amount of space that should be added after a character 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 character 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 character *c1* appears next to character *c2* the space between them should be increased by *n*. Most entries in kernpairs section will have a negative value for *n*.

## FILES

**/usr/share/groff/1.18.1.4/font/dev***name***/DESC** Device description file for device *name*.

**/usr/share/groff/1.18.1.4/font/dev***name*/*F* Font file for font *F* of device *name*.

## SEE ALSO

**groff_out**(5), **troff**(1).