Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'groff.1' command

$ man groff.1

GROFF(1) General Commands Manual GROFF(1)

NAME

groff - front-end for the groff document formatting system

SYNOPSIS

groff [-abcegijklpstzCEGNRSUVXZ] [-d cs] [-D arg] [-f fam] [-F dir]

[-I dir] [-K arg] [-L arg] [-m name] [-M dir] [-n num] [-o list]

[-P arg] [-r cn] [-T dev] [-w name] [-W name] [file ...]

groff -h

groff --help

groff -v [option ...]

groff --version [option ...]

DESCRIPTION

This document describes the groff program, the main front-end for the groff document formatting system. The groff program and macro suite is the implementation of a roff(7) system within the free software collec? tion GNU ?http://www.gnu.org?. The groff system has all features of the classical roff, but adds many extensions.

The groff program allows control of the whole groff system by command-line options. This is a great simplification in comparison to the classical case (which uses pipes only).

OPTIONS

The command line is parsed according to the usual GNU convention. Whitespace is permitted between a command-line option and its argument.

Options can be grouped behind a single ?-? (minus character). A file? name of - (minus character) denotes the standard input.

As groff is a wrapper program for troff both programs share a set of options. But the groff program has some additional, native options and gives a new meaning to some troff options. On the other hand, not all troff options can be fed into groff.

Native groff Options

The following options either do not exist for troff or are differently interpreted by groff.

-D arg Set default input encoding used by preconv to arg. Implies -k.

-e Preprocess with eqn.

-g Preprocess with grn.

-G Preprocess with grap. Implies -p.

-h

--help Print a help message.

-I dir This option may be used to specify a directory to search for files (both those on the command line and those named in .psbb and .so requests, and \X'ps: import' , \X'ps: file' and \X'pdf: pdfpic' escapes). The current directory is always searched first. This option may be specified more than once; the direc? tories are searched in the order specified. No directory search is performed for files specified using an absolute path. This option implies the -s option.

-j Preprocess with chem. Implies -p.

-k Preprocess with preconv. This is run before any other pre? processor. Please refer to preconv's manual page for its behav? iour if no -K (or -D) option is specified.

-K arg Set input encoding used by preconv to arg. Implies -k.

-l Send the output to a spooler program for printing. The command that should be used for this is specified by the print command in the device description file, see groff_font(5). If this com? mand is not present, the output is piped into the lpr(1) program by default. See options -L and -X.

-L arg Pass arg to the spooler program. Several arguments should be passed with a separate -L option each. Note that groff does not prepend ?-? (a minus sign) to arg before passing it to the spooler program.

-N Don't allow newlines within eqn delimiters. This is the same as the -N option in eqn.

-p Preprocess with pic.

-P -option

-P -option -P arg

Pass -option or -option arg to the postprocessor. The option must be specified with the necessary preceding minus sign(s) ?-? or ?--? because groff does not prepend any dashes before passing it to the postprocessor. For example, to pass a title to the gxditview postprocessor, the shell command

groff -X -P -title -P 'groff it' foo

is equivalent to

groff -X -Z foo | gxditview -title 'groff it' -

-R Preprocess with refer. No mechanism is provided for passing ar? guments to refer because most refer options have equivalent lan? guage elements that can be specified within the document. See refer(1) for more details.

-s Preprocess with soelim.

-S Safer mode. Pass the -S option to pic and disable the following troff requests: .open, .opena, .pso, .sy, and .pi. For security reasons, safer mode is enabled by default.

-t Preprocess with tbl.

-T dev Set output device to dev. For this device, troff generates the intermediate output; see groff_out(5). Then groff calls a post? processor to convert troff's intermediate output to its final format. Real devices in groff are

dvi TeX DVI format (postprocessor is grodvi).

html

xhtml HTML and XHTML output (preprocessors are soelim

and pre-grohtml, postprocessor is post-grohtml).

lbp Canon CAPSL printers (LBP-4 and LBP-8 series laser printers; postprocessor is grolbp).

lj4 HP LaserJet4 compatible (or other PCL5 compatible) printers (postprocessor is grolj4).

ps PostScript output (postprocessor is grops).

pdf Portable Document Format (PDF) output (postproces? sor is gropdf).

For the following TTY output devices (postprocessor is always grotty), -T selects the output encoding:

ascii 7bit ASCII.

cp1047 Latin-1 character set for EBCDIC hosts.

latin1 ISO 8859-1.

utf8 Unicode character set in UTF-8 encoding. This mode has the most useful fonts for TTY mode, so it is the best mode for TTY output.

The following arguments select gxditview as the ?postprocessor? (it is rather a viewing program):

X75 75dpi resolution, 10pt document base font.

X75-12 75dpi resolution, 12pt document base font.

X100 100dpi resolution, 10pt document base font.

X100-12

100dpi resolution, 12pt document base font.

The default device is ps.

-U Unsafe mode. Reverts to the (old) unsafe behaviour; see option -S.

-v

--version

Output version information of groff and of all programs that are run by it; that is, the given command line is parsed in the usual way, passing -v to all subprograms.

-V Output the pipeline that would be run by groff (as a wrapper program) on the standard output, but do not execute it. If

given more than once, the commands are both printed on the stan? dard error and run.

-X Use gxditview instead of using the usual postprocessor to (pre)view a document. The printing spooler behavior as outlined with options -l and -L is carried over to gxditview(1) by deter? mining an argument for the -printCommand option of gxditview(1). This sets the default Print action and the corresponding menu entry to that value. -X only produces good results with -Tps, -TX75, -TX75-12, -TX100, and -TX100-12. The default resolution for previewing -Tps output is 75dpi; this can be changed by passing the -resolution option to gxditview, for example

groff -X -P-resolution -P100 -man foo.1

-z Suppress output generated by troff. Only error messages are printed.

-Z Do not automatically postprocess groff intermediate output in the usual manner. This will cause the troff output to appear on standard output, replacing the usual postprocessor output; see groff_out(5).

Transparent Options

The following options are transparently handed over to the formatter program troff that is called by groff subsequently. These options are described in more detail in troff(1).

-a ASCII approximation of output.

-b Backtrace on error or warning.

-c Disable color output. Please consult the grotty(1) man page for more details.

-C Enable compatibility mode.

-d cs

-d name=s

Define string.

-E Disable troff error messages.

-f fam Set default font family.

-F dir Set path for device DESC files.

-i Process standard input after the specified input files.

-m name

Include macro file name.tmac (or tmac.name); see also groff_tmac(5).

-M dir Path for macro files.

-n num Number the first page num.

-o list

Output only pages in list.

-r cn

-r name=n

Set number register.

-w name

Enable warning name. See troff(1) for names.

-W name

disable warning name. See troff(1) for names.

USING GROFF

The groff system implements the infrastructure of classical roff; see roff(7) for a survey on how a roff system works in general. Due to the front-end programs available within the groff system, using groff is much easier than classical roff. This section gives an overview of the parts that constitute the groff system. It complements roff(7) with groff-specific features. This section can be regarded as a guide to the documentation around the groff system.

Paper Size

The virtual paper size used by troff to format the input is controlled globally with the requests .po, .pl, and .ll. See groff_tmac(5) for the ?papersize? macro package which provides a convenient interface. The physical paper size, giving the actual dimensions of the paper sheets, is controlled by output devices like grops with the command-line options -p and -l. See groff_font(5) and the man pages of the output devices for more details. groff uses the command-line option -P to pass options to output devices; for example, the following selects A4 paper in landscape orientation for the PS device:

groff -Tps -P-pa4 -P-l ...

Front-ends

The groff program is a wrapper around the troff(1) program. It allows one to specify the preprocessors by command-line options and automatically runs the postprocessor that is appropriate for the selected device. Doing so, the sometimes tedious piping mechanism of classical roff(7) can be avoided.

The grog(1) program can be used for guessing the correct groff command line to format a file.

The groffer(1) program is an all-around viewer for groff files and man pages.

Preprocessors

The groff preprocessors are reimplementations of the classical preprocessors with moderate extensions. The standard preprocessors distributed with the groff package are

eqn(1) for mathematical formulae,

grn(1) for including gremlin(1) pictures,

pic(1) for drawing diagrams,

chem(1)
for chemical structure diagrams,

refer(1)
for bibliographic references,

soelim(1)
for including macro files from standard locations,

and

tbl(1) for tables.

A new preprocessor not available in classical troff is preconv(1) which converts various input encodings to something groff can understand. It is always run first before any other preprocessor.

Besides these, there are some internal preprocessors that are automatically run with some devices. These aren't visible to the user.

Macro Packages

Macro packages can be included by option -m. The groff system imple?

ments and extends all classical macro packages in a compatible way and adds some packages of its own. Actually, the following macro packages come with groff:

man The traditional man page format; see groff_man(7). It can be specified on the command line as -man or -m man.

mandoc The general package for man pages; it automatically recognizes whether the documents uses the man or the mdoc format and branches to the corresponding macro package. It can be speci? fied on the command line as -mandoc or -m mandoc.

mdoc The BSD-style man page format; see groff_mdoc(7). It can be specified on the command line as -mdoc or -m mdoc.

me The classical me document format; see groff_me(7). It can be specified on the command line as -me or -m me.

mm The classical mm document format; see groff_mm(7). It can be specified on the command line as -mm or -m mm.

ms The classical ms document format; see groff_ms(7). It can be specified on the command line as -ms or -m ms.

www HTML-like macros for inclusion in arbitrary groff documents; see groff_www(7).

Details on the naming of macro files and their placement can be found in groff_tmac(5); this man page also documents some other, minor auxil? iary macro packages not mentioned here.

Programming Language

General concepts common to all roff programming languages are described in roff(7).

The groff extensions to the classical troff language are documented in groff_diff(7).

An overview of language features, including all supported escapes and requests, can be found in groff(7).

Formatters

The central roff formatter within the groff system is troff(1). It provides the features of both the classical troff and nroff, as well as the groff extensions. The command-line option -C switches troff into

compatibility mode which tries to emulate classical roff as much as possible.

There is a shell script nroff(1) that emulates the behavior of classi? cal nroff. It tries to automatically select the proper output encod? ing, according to the current locale.

The formatter program generates intermediate output; see groff_out(7).

Devices

In roff, the output targets are called devices. A device can be a piece of hardware, e.g., a printer, or a software file format. A de? vice is specified by the option -T. The groff devices are as follows.

ascii Text output using the ascii(7) character set.

cp1047 Text output using the EBCDIC code page IBM cp1047 (e.g., OS/390 Unix).

dvi TeX DVI format.

html HTML output.

latin1 Text output using the ISO Latin-1 (ISO 8859-1) character set; see iso_8859_1(7).

lbp Output for Canon CAPSL printers (LBP-4 and LBP-8 series laser printers).

lj4 HP LaserJet4-compatible (or other PCL5-compatible) printers.

ps PostScript output; suitable for printers and previewers like gv(1).

pdf PDF files; suitable for viewing with tools such as evince(1) and okular(1).

utf8 Text output using the Unicode (ISO 10646) character set with UTF-8 encoding; see unicode(7).

xhtml XHTML output.

X75 75dpi X Window System output suitable for the previewers xditview(1x) and gxditview(1). A variant for a 12pt document base font is X75-12.

X100 100dpi X Window System output suitable for the previewers xditview(1x) and gxditview(1). A variant for a 12pt document base font is X100-12.

The postprocessor to be used for a device is specified by the postpro command in the device description file; see groff_font(5). This can be overridden with the -X option.

The default device is ps.

Postprocessors

groff provides 3 hardware postprocessors:

grolbp(1)
: for some Canon printers,

grolj4(1)
: for printers compatible to the HP LaserJet 4 and PCL5,

grotty(1)
: for text output using various encodings, e.g., on text-oriented terminals or line printers.

Today, most printing or drawing hardware is handled by the operating system, by device drivers, or by software interfaces, usually accepting PostScript. Consequently, there isn't an urgent need for more hardware device postprocessors.

The groff software devices for conversion into other document file for? mats are

grodvi(1)
: for the DVI format,

grohtml(1)
: for HTML and XHTML formats,

grops(1)
: for PostScript.

gropdf(1)
: for PDF.

Combined with the many existing free conversion tools this should be sufficient to convert a troff document into virtually any existing data format.

Utilities

The following utility programs around groff are available.

addftinfo(1)

Add information to troff font description files for use with groff.

afmtodit(1)

Create font description files for PostScript device.

eqn2graph(1)

Convert an eqn image into a cropped image.

gdiffmk(1)

Mark differences between groff, nroff, or troff files.

grap2graph(1)

Convert a grap diagram into a cropped bitmap image.

groffer(1)

General viewer program for groff files and man pages.

gxditview(1)

The groff X viewer, the GNU version of xditview.

hpftodit(1)

Create font description files for lj4 device.

indxbib(1)

Make inverted index for bibliographic databases.

lkbib(1)

Search bibliographic databases.

lookbib(1)

Interactively search bibliographic databases.

pdfroff(1)

Create PDF documents using groff.

pfbtops(1)

Translate a PostScript font in .pfb format to ASCII.

pic2graph(1)

Convert a pic diagram into a cropped image.

tfmtodit(1)

Create font description files for TeX DVI device.

xditview(1x)

roff viewer historically distributed with the X Window System.

xtotroff(1)

Convert X font metrics into GNU troff font metrics.

ENVIRONMENT

Normally, the path separator in the following environment variables is the colon; this may vary depending on the operating system. For exam? ple, DOS and Windows use a semicolon instead.

GROFF_BIN_PATH

This search path, followed by PATH, is used for commands that are executed by groff. If it is not set then the directory where the groff binaries were installed is prepended to PATH.

GROFF_COMMAND_PREFIX

When there is a need to run different roff implementations at the same time groff provides the facility to prepend a prefix to most of its programs that could provoke name clashings at run time (default is to have none). Historically, this prefix was the character g, but it can be anything. For example, gtroff stood for groff's troff, gtbl for the groff version of tbl. By setting GROFF_COMMAND_PREFIX to different values, the different roff installations can be addressed. More exactly, if it is set to prefix xxx then groff as a wrapper program internally calls xxxtroff instead of troff. This also applies to the preproces? sors eqn, grn, pic, refer, tbl, soelim, and to the utilities indxbib and lookbib. This feature does not apply to any pro? grams different from the ones above (most notably groff itself) since they are unique to the groff package.

GROFF_ENCODING

The value of this environment value is passed to the preconv preprocessor to select the encoding of input files. Setting this option implies groff's command-line option -k (this is, groff actually always calls preconv). If set without a value, groff calls preconv without arguments. An explicit -K command-line option overrides the value of GROFF_ENCODING. See pre? conv(1) for details.

GROFF_FONT_PATH

A list of directories in which to search for the devname direc?
tory in addition to the default ones. See troff(1) and
groff_font(5) for more details.

GROFF_TMAC_PATH

A list of directories in which to search for macro files in ad?
dition to the default directories. See troff(1) and
groff_tmac(5) for more details.

GROFF_TMPDIR

The directory in which temporary files are created. If this is
not set but the environment variable TMPDIR instead, temporary
files are created in the directory TMPDIR. On MS-DOS and Win?
dows platforms, the environment variables TMP and TEMP (in that
order) are searched also, after GROFF_TMPDIR and TMPDIR. Other?
wise, temporary files are created in /tmp. The refer(1),
groffer(1), grohtml(1), and grops(1) commands use temporary
files.

GROFF_TYPESETTER

Preset the default device. If this is not set the ps device is
used as default. This device name is overwritten by the option
-T.

EXAMPLES

The following example illustrates the power of the groff program as a
wrapper around troff.

To process a roff file using the preprocessors tbl and pic and the me
macro set, classical troff had to be called by

```
pic foo.me | tbl | troff -me -Tlatin1 | grotty
```

Using groff, this pipe can be shortened to the equivalent command

```
groff -p -t -me -T latin1 foo.me
```

An even easier way to call this is to use grog(1) to guess the pre?
processor and macro options and execute the generated command (by using
backquotes to specify shell command substitution)

```
`grog -Tlatin1 foo.me`
```

The simplest way is to view the contents in an automated way by calling

groffer foo.me

BUGS

On EBCDIC hosts (e.g., OS/390 Unix), output devices ascii and latin1 aren't available. Similarly, output for EBCDIC code page cp1047 is not available on ASCII based operating systems.

INSTALLATION DIRECTORIES

groff installs files in varying locations depending on its compile-time configuration. On this installation, the following locations are used.

/usr/share/X11/app-defaults

Application defaults directory for gxditview(1).

/usr/bin

Directory containing groff's executable commands.

/usr/share/groff/1.22.4/eign

List of common words for indxbib(1).

/usr/share/groff/1.22.4

Directory for data files.

/usr/dict/papers/Ind

Default index for lkbib(1) and refer(1).

/usr/share/doc/groff

Documentation directory.

/usr/share/doc/groff/examples

Example directory.

/usr/share/groff/1.22.4/font

Font directory.

/usr/share/doc/groff/html

HTML documentation directory.

/usr/lib/font

Legacy font directory.

/etc/groff/site-font

Local font directory.

/etc/groff/site-tmac

Local macro package (tmac file) directory.

/usr/share/groff/1.22.4/tmac

Macro package (tmac file) directory.

/usr/share/groff/1.22.4/oldfont

Font directory for compatibility with old versions of groff; see grops(1).

/usr/share/doc/groff/pdf

PDF documentation directory.

/etc/groff/site-tmac

System macro package (tmac file) directory.

groff Macro Directory

This contains all information related to macro packages. Note that more than a single directory is searched for those files as documented in groff_tmac(5). For the groff installation corresponding to this document, it is located at /usr/share/groff/1.22.4/tmac. The following files contained in the groff macro directory have a special meaning:

troffrc

Initialization file for troff. This is interpreted by troff be? fore reading the macro sets and any input.

troffrc-end

Final startup file for troff. It is parsed after all macro sets have been read.

name.tmac

tmac.name

Macro file for macro package name.

groff Font Directory

This contains all information related to output devices. Note that more than a single directory is searched for those files; see troff(1). For the groff installation corresponding to this document, it is lo? cated at /usr/share/groff/1.22.4/font. The following files contained in the groff font directory have a special meaning:

devname/DESC

Device description file for device name, see groff_font(5).

devname/F

Font file for font F of device name.

AVAILABILITY

Information on how to get groff and related information is available at the groff page of the GNU website ?http://www.gnu.org/software/groff?.

Three groff mailing lists are available:

bug tracker activity (read-only) ?bug-groff@gnu.org?;

general discussion ?groff@gnu.org?; and

commit activity (read-only) ?groff-commit@gnu.org?, which re? ports changes to groff's source code repository by its develop? ers.

Details on repository access and much more can be found in the file README at the top directory of the groff source package.

A free implementation of the grap preprocessor, written by Ted Faber ?faber@lunabase.org?, can be found at the grap website ?http://www.lunabase.org/~faber/Vault/software/grap/?. This is the only grap supported by groff.

AUTHORS

groff was written by James Clark ?jjc@jclark.com?. This document was rewritten, enhanced, and put under the FDL license in 2002 by Bernd Warken ?groff-bernd.warken-72@web.de?.

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?.

Due to its complex structure, the groff system has many man pages. They can be read with man(1) or groffer(1).

But there are special sections of man pages. groff has man pages in sections 1, 5,and 7. When there are several man pages with the same name in the same man section, the one with the lowest section is should as first. The other man pages can be shown anyway by adding the sec? tion number as argument before the man page name. Reading the man page about the groff language is done by one of

man 7 groff

groffer 7 groff

Introduction, history and further readings:

roff(7).

Viewer for groff files:

groffer(1), gxditview(1), xditview(1x).

Wrapper programs for formatters:

groff(1), grog(1).

Roff preprocessors:

eqn(1), grn(1), pic(1), chem(1), preconv(1), refer(1),

soelim(1), tbl(1), grap(1).

Roff language with the groff extensions:

groff(7), groff_char(7), groff_diff(7), groff_font(5).

Roff formatter programs:

nroff(1), troff(1), ditroff(7).

The intermediate output language:

groff_out(7).

Postprocessors for the output devices:

grodvi(1), grohtml(1), grolbp(1), grolj4(1), lj4_font(5),

grops(1), gropdf(1), grotty(1).

Groff macro packages and macro-specific utilities:

groff_tmac(5), groff_man(7), groff_mdoc(7), groff_me(7),

groff_mm(7), groff_mmse(7), groff_mom(7), groff_ms(7),

groff_www(7), groff_trace(7), mmroff(7).

The following utilities are available:

addftinfo(1), afmtodit(1), eqn2graph(1), gdiffmk(1),

grap2graph(1), groffer(1), gxditview(1), hpftodit(1),

indxbib(1), lkbib(1), lookbib(1), pdfroff(1), pfbtops(1),

pic2graph(1), tfmtodit(1), xtotroff(1).