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 'grops.1' command

$ man grops.1

GROPS(1) General Commands Manual GROPS(1)

NAME

grops - PostScript driver for groff

SYNOPSIS

grops [-glmv] [-b n] [-c n] [-F dir] [-I dir] [-p papersize]

[-P prologue] [-w n] [file ...]

DESCRIPTION

grops translates the output of GNU troff to PostScript. Normally grops should be invoked by using the groff command with a -Tps option. (Ac? tually, this is the default for groff.) If no files are given, grops reads the standard input. A filename of - also causes grops to read the standard input. PostScript output is written to the standard out? put. When grops is run by groff options can be passed to grops using groff's -P option.

Note that grops doesn't produce a valid document structure (conforming to the Document Structuring Convention) if called with multiple file arguments. To print such concatenated output it is necessary to deac? tivate DSC handling in the printing program or previewer. See section ?Font Installation? below for a guide how to install fonts for grops.

OPTIONS

Whitespace is permitted between a command-line option and its argument.

-bn Provide workarounds for older printers, broken spoolers, and previewers. Normally grops produces output at PostScript Lan?

guageLevel 2 that conforms to the Document Structuring Conven?
tions version 3.0. Some older printers, spoolers, and preview?
ers can't handle such output. The value of n controls what
grops does to make its output acceptable to such programs. A
value of 0 causes grops not to employ any workarounds.
Add 1 if no %%BeginDocumentSetup and %%EndDocumentSetup comments
should be generated; this is needed for early versions of Tran?
Script that get confused by anything between the %%EndProlog
comment and the first %%Page comment.
Add 2 if lines in included files beginning with %! should be
stripped out; this is needed for Sun's pageview previewer.
Add 4 if %%Page, %%Trailer and %%EndProlog comments should be
stripped out of included files; this is needed for spoolers that
don't understand the %%BeginDocument and %%EndDocument comments.
Add 8 if the first line of the PostScript output should be %!PS-
Adobe-2.0 rather than %!PS-Adobe-3.0; this is needed when using
Sun's Newsprint with a printer that requires page reversal.
Add 16 if no media size information should be included in the
document (this is, neither use %%DocumentMedia nor the set?
pagedevice PostScript command). This was the behaviour of groff
version 1.18.1 and earlier; it is needed for older printers
which don't understand PostScript LanguageLevel 2. It is also
necessary if the output is further processed to get an encapsu?
lated PS (EPS) file ? see below.
The default value can be specified by a

broken n

command in the DESC file. Otherwise the default value is 0.

-cn Print n copies of each page.

-Fdir Prepend directory dir/devname to the search path for prologue,
font, and device description files; name is the name of the de?
vice, usually ps.

-g Guess the page length. This generates PostScript code that
guesses the page length. The guess is correct only if the im?

ageable area is vertically centered on the page. This option allows you to generate documents that can be printed both on letter (8.5?11) paper and on A4 paper without change.

-Idir This option may be used to add a directory to the search path for files on the command line and files named in \X'ps: import' and \X'ps: file' escapes. The search path is initialized with the current directory. This option may be specified more than once; the directories are then searched in the order specified (but before the current directory). If you want to make the current directory be read before other directories, add -I. at the appropriate place.

No directory search is performed for files with an absolute file name.

-l Print the document in landscape format.

-m Turn manual feed on for the document.

-ppaper-size

Set physical dimension of output medium. This overrides the pa? persize, paperlength, and paperwidth commands in the DESC file; it accepts the same arguments as the papersize command. See groff_font (5) for details.

-Pprologue-file

Use the file prologue-file (in the font path) as the prologue instead of the default prologue file prologue. This option overrides the environment variable GROPS_PROLOGUE.

-wn Lines should be drawn using a thickness of n thousandths of an em. If this option is not given, the line thickness defaults to 0.04 em.

-v Print the version number.

USAGE

The input to grops must be in the format output by troff(1). This is described in groff_out(5).

In addition, the device and font description files for the device used must meet certain requirements: The resolution must be an integer mul?

tiple of 72 times the sizescale. The ps device uses a resolution of 72000 and a sizescale of 1000.

The device description file must contain a valid paper size; see groff_font(5) for more information.

Each font description file must contain a command

internalname psname

which says that the PostScript name of the font is psname. It may also contain a command

encoding enc_file

which says that the PostScript font should be reencoded using the en? coding described in enc_file; this file should consist of a sequence of lines of the form:

pschar code

where pschar is the PostScript name of the character, and code is its position in the encoding expressed as a decimal integer; valid values are in the range 0 to 255. Lines starting with # and blank lines are ignored. The code for each character given in the font file must cor? respond to the code for the character in encoding file, or to the code in the default encoding for the font if the PostScript font is not to be reencoded. This code can be used with the \N escape sequence in troff to select the character, even if the character does not have a groff name. Every character in the font file must exist in the Post? Script font, and the widths given in the font file must match the widths used in the PostScript font. grops assumes that a character with a groff name of space is blank (makes no marks on the page); it can make use of such a character to generate more efficient and compact PostScript output.

Note that grops is able to display all glyphs in a PostScript font, not only 256. enc_file (or the default encoding if no encoding file speci? fied) just defines the order of glyphs for the first 256 characters; all other glyphs are accessed with additional encoding vectors which grops produces on the fly.

grops can automatically include the downloadable fonts necessary to

print the document. Such fonts must be in PFA format. Use pfbtops(1) to convert a Type 1 font in PFB format. Any downloadable fonts which should, when required, be included by grops must be listed in the file /usr/share/groff/1.22.4/font/devps/download; this should consist of lines of the form

font filename

where font is the PostScript name of the font, and filename is the name of the file containing the font; lines beginning with # and blank lines are ignored; fields may be separated by tabs or spaces; filename is searched for using the same mechanism that is used for groff font met? ric files. The download file itself is also searched for using this mechanism; currently, only the first found file in the font path is used.

If the file containing a downloadable font or imported document con? forms to the Adobe Document Structuring Conventions, then grops inter? prets any comments in the files sufficiently to ensure that its own output is conforming. It also supplies any needed font resources that are listed in the download file as well as any needed file resources. It is also able to handle inter-resource dependencies. For example, suppose that you have a downloadable font called Garamond, and also a downloadable font called Garamond-Outline which depends on Garamond (typically it would be defined to copy Garamond's font dictionary, and change the PaintType), then it is necessary for Garamond to appear be? fore Garamond-Outline in the PostScript document. grops handles this automatically provided that the downloadable font file for Garamond-Outline indicates its dependence on Garamond by means of the Document Structuring Conventions, for example by beginning with the following lines

```
%!PS-Adobe-3.0 Resource-Font
%%DocumentNeededResources: font Garamond
%%EndComments
%%IncludeResource: font Garamond
```

In this case both Garamond and Garamond-Outline would need to be listed

in the download file. A downloadable font should not include its own name in a %%DocumentSuppliedResources comment.

grops does not interpret %%DocumentFonts comments. The %%Document?NeededResources, %%DocumentSuppliedResources, %%IncludeResource, %%BeginResource, and %%EndResource comments (or possibly the old %%DocumentNeededFonts, %%DocumentSuppliedFonts, %%IncludeFont, %%Begin?Font, and %%EndFont comments) should be used.

In the default setup there are styles called R, I, B, and BI mounted at font positions 1 to 4. The fonts are grouped into families A, BM, C, H, HN, N, P, and T having members in each of these styles:

AR AvantGarde-Book

AI AvantGarde-BookOblique

AB AvantGarde-Demi

ABI AvantGarde-DemiOblique

BMR Bookman-Light

BMI Bookman-LightItalic

BMB Bookman-Demi

BMBI Bookman-DemiItalic

CR Courier

CI Courier-Oblique

CB Courier-Bold

CBI Courier-BoldOblique

HR Helvetica

HI Helvetica-Oblique

HB Helvetica-Bold

HBI Helvetica-BoldOblique

HNR Helvetica-Narrow

HNI Helvetica-Narrow-Oblique

HNB Helvetica-Narrow-Bold

HNBI Helvetica-Narrow-BoldOblique

NR NewCenturySchlbk-Roman

NI NewCenturySchlbk-Italic

NB NewCenturySchlbk-Bold

NBI NewCenturySchlbk-BoldItalic

PR Palatino-Roman

PI Palatino-Italic

PB Palatino-Bold

PBI Palatino-BoldItalic

TR Times-Roman

TI Times-Italic

TB Times-Bold

TBI Times-BoldItalic

There is also the following font which is not a member of a family:

ZCMI ZapfChancery-MediumItalic

There are also some special fonts called S for the PS Symbol font, and SS, containing slanted lowercase Greek letters taken from PS Symbol. Zapf Dingbats is available as ZD, and a reversed version of ZapfDing? bats (with symbols pointing in the opposite direction) is available as ZDR; most characters in these fonts are unnamed and must be accessed using \N.

The default color for \m and \M is black; for colors defined in the ?rgb? color space setrgbcolor is used, for ?cmy? and ?cmyk? setcmyk? color, and for ?gray? setgray. Note that setcmykcolor is a PostScript LanguageLevel 2 command and thus not available on some older printers. grops understands various X commands produced using the \X escape se? quence; grops only interprets commands that begin with a ps: tag.

\X'ps: exec code'

This executes the arbitrary PostScript commands in code. The PostScript currentpoint is set to the position of the \X command before executing code. The origin is at the top left corner of the page, and y coordinates increase down the page. A proce? dure u is defined that converts groff units to the coordinate system in effect (provided the user doesn't change the scale).

For example,

.nr x 1i

\X'ps: exec \nx u 0 rlineto stroke'

draws a horizontal line one inch long. code may make changes to the graphics state, but any changes persist only to the end of the page. A dictionary containing the definitions specified by the def and mdef is on top of the dictionary stack. If your code adds definitions to this dictionary, you should allocate space for them using \X'ps mdef n'. Any definitions persist only until the end of the page. If you use the \Y escape se? quence with an argument that names a macro, code can extend over multiple lines. For example,

```
.nr x 1i
.de y
ps: exec
\nx u 0 rlineto
stroke
..
\Yy
```

is another way to draw a horizontal line one inch long. Note the single backslash before ?nx? ? the only reason to use a num? ber register while defining the macro ?y? is to convert a user-specified dimension ?1i? to internal groff units which are in turn converted to PS units with the u procedure.

grops wraps user-specified PostScript code into a dictionary, nothing more. In particular, it doesn't start and end the in? serted code with save and restore, respectively. This must be supplied by the user, if necessary.

\X'ps: file name'

This is the same as the exec command except that the PostScript code is read from file name.

\X'ps: def code'

Place a PostScript definition contained in code in the prologue. There should be at most one definition per \X command. Long definitions can be split over several \X commands; all the code arguments are simply joined together separated by newlines. The

definitions are placed in a dictionary which is automatically pushed on the dictionary stack when an exec command is executed.

If you use the \Y escape sequence with an argument that names a macro, code can extend over multiple lines.

\X'ps: mdef n code'

Like def, except that code may contain up to n definitions. grops needs to know how many definitions code contains so that it can create an appropriately sized PostScript dictionary to contain them.

\X'ps: import file llx lly urx ury width [height]'

Import a PostScript graphic from file. The arguments llx, lly, urx, and ury give the bounding box of the graphic in the default PostScript coordinate system; they should all be integers; llx and lly are the x and y coordinates of the lower left corner of the graphic; urx and ury are the x and y coordinates of the up? per right corner of the graphic; width and height are integers that give the desired width and height in groff units of the graphic.

The graphic is scaled so that it has this width and height and translated so that the lower left corner of the graphic is lo? cated at the position associated with \X command. If the height argument is omitted it is scaled uniformly in the x and y direc? tions so that it has the specified width.

Note that the contents of the \X command are not interpreted by troff; so vertical space for the graphic is not automatically added, and the width and height arguments are not allowed to have attached scaling indicators.

If the PostScript file complies with the Adobe Document Struc? turing Conventions and contains a %%BoundingBox comment, then the bounding box can be automatically extracted from within groff by using the psbb request.

See groff_tmac(5) for a description of the PSPIC macro which provides a convenient high-level interface for inclusion of

PostScript graphics.

\X'ps: invis'

\X'ps: endinvis'

No output is generated for text and drawing commands that are bracketed with these \X commands. These commands are intended for use when output from troff is previewed before being pro? cessed with grops; if the previewer is unable to display certain characters or other constructs, then other substitute characters or constructs can be used for previewing by bracketing them with these \X commands.

For example, gxditview is not able to display a proper \(em character because the standard X11 fonts do not provide it; this problem can be overcome by executing the following request

```
.char \(em \X'ps: invis'\
\Z'\v'-.25m'\h'.05m'\D'l .9m 0'\h'.05m''\
\X'ps: endinvis'\(em
```

In this case, gxditview is unable to display the \(em character and draws the line, whereas grops prints the \(em character and ignores the line (this code is already in file Xps.tmac which is loaded if a document intended for grops is previewed with gxditview).

If a PostScript procedure BPhook has been defined via a ?ps: def? or ?ps: mdef? device command, it is executed at the beginning of every page (before anything is drawn or written by groff). For example, to underlay the page contents with the word ?DRAFT? in light gray, you might use

```
.de XX
ps: def
/BPhook
{ gsave .9 setgray clippath pathbbox exch 2 copy
  .5 mul exch .5 mul translate atan rotate pop pop
  /NewCenturySchlbk-Roman findfont 200 scalefont setfont
  (DRAFT) dup stringwidth pop -.5 mul -70 moveto show
```

```
  grestore }
def
..
.devicem XX
```

Or, to cause lines and polygons to be drawn with square linecaps and mitered linejoins instead of the round linecaps and linejoins normally used by grops, use

```
.de XX
ps: def
/BPhook { 2 setlinecap 0 setlinejoin } def
..
.devicem XX
```

(square linecaps, as opposed to butt linecaps (0 setlinecap), give true corners in boxed tables even though the lines are drawn unconnected).

Encapsulated PostScript

grops itself doesn't emit bounding box information. With the help of Ghostscript the following simple script, groff2eps, produces an encap? sulated PS file.

```
#! /bin/sh
groff -P-b16 $1 > $1.ps
gs -dNOPAUSE -sDEVICE=bbox -- $1.ps 2> $1.bbox
sed -e "/^%%Orientation/r $1.bbox" \
    -e "/^%!PS-Adobe-3.0/s/$/ EPSF-3.0/" $1.ps > $1.eps
rm $1.ps $1.bbox
```

Just say

```
groff2eps foo
```

to convert file foo to foo.eps.

TrueType and other font formats

TrueType fonts can be used with grops if converted first to Type 42 format, a special PostScript wrapper equivalent to the PFA format men? tioned in pfbtops(1). There are several different methods to generate a type42 wrapper and most of them involve the use of a PostScript in? terpreter such as Ghostscript ? see gs(1).

Yet, the easiest method involves the use of the application ttftot42(1). This program uses freetype(3) (version 1.3.1) to generate type42 font wrappers and well-formed AFM files that can be fed to the afmtodit(1) script to create appropriate metric files. The resulting font wrappers should be added to the download file. ttftot42 source code can be downloaded from ftp://www.giga.or.at/pub/nih/ttftot42/ ?ftp://www.giga.or.at/pub/nih/ttftot42/?.

Another solution for creating type42 wrappers is to use FontForge, available from http://fontforge.sf.net ?http://fontforge.sf.net?. This font editor can convert most outline font formats.

FONT INSTALLATION

This section gives a summary of the above explanations; it can serve as a step-by-step font installation guide for grops.

? Convert your font to something groff understands. This is either a PostScript Type 1 font in PFA format or a PostScript Type 42 font, together with an AFM file.

The very first characters in a PFA file look like this:

```
%!PS-AdobeFont-1.0:
```

A PFB file has this also in the first line, but the string is pre? ceded with some binary bytes.

The very first characters in a Type 42 font file look like this:

```
%!PS-TrueTypeFont
```

This is a wrapper format for TrueType fonts. Old PS printers might not support it (this is, they don't have a built-in TrueType font interpreter).

If your font is in PFB format (such fonts normally have .pfb as the file extension), you might use groff's pfbtops(1) program to con? vert it to PFA. For TrueType fonts, try ttftot42 or fontforge.

For all other font formats use fontforge which can convert most outline font formats.

? Convert the AFM file to a groff font description file with the afmtodit(1) program. An example call is

```
afmtodit Foo-Bar-Bold.afm textmap FBB
```

which converts the metric file Foo-Bar-Bold.afm to the groff font FBB. If you have a font family which comes with normal, bold, italic, and bold italic faces, it is recommended to use the letters R, B, I, and BI, respectively, as postfixes in the groff font names to make groff's ?.fam? request work. An example is groff's built-in Times-Roman font: The font family name is T, and the groff font names are TR, TB, TI, and TBI.

? Install both the groff font description files and the fonts in a devps subdirectory of the font path which groff finds. See section ?Environment? in troff(1) for the actual value of the font path. Note that groff doesn't use the AFM files (but it is a good idea to store them anyway).

? Register all fonts which must be downloaded to the printer in the devps/download file. Only the first occurrence of this file in the font path is read. This means that you should copy the default download file to the first directory in your font path and add your fonts there. To continue the above example we assume that the PS font name for Foo-Bar-Bold.pfa is ?XY-Foo-Bar-Bold? (the PS font name is stored in the internalname field in the FBB file), thus the following line should be added to download.

```
XY-Foo-Bar-Bold Foo-Bar-Bold.pfa
```

OLD FONTS

groff versions 1.19.2 and earlier contain a slightly different set of the 35 Adobe core fonts; the difference is mainly the lack of the ?Euro? glyph and a reduced set of kerning pairs. For backwards compat?ibility, these old fonts are installed also in the

```
/usr/share/groff/1.22.4/oldfont/devps
```

directory.

To use them, make sure that grops finds the fonts before the default system fonts (with the same names): Either add command-line option -F to grops

```
groff -Tps -P-F -P/usr/share/groff/1.22.4/oldfont ...
```

or add the directory to groff's font path environment variable

GROFF_FONT_PATH=/usr/share/groff/1.22.4/oldfont

ENVIRONMENT

GROPS_PROLOGUE

If this is set to foo, then grops uses the file foo (in the font path) instead of the default prologue file prologue. The option -P overrides this environment variable.

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.

SOURCE_DATE_EPOCH

A timestamp (expressed as seconds since the Unix epoch) to use as the creation timestamp in place of the current time.

FILES

/usr/share/groff/1.22.4/font/devps/DESC

Device description file.

/usr/share/groff/1.22.4/font/devps/F

Font description file for font F.

/usr/share/groff/1.22.4/font/devps/download

List of downloadable fonts.

/usr/share/groff/1.22.4/font/devps/text.enc

Encoding used for text fonts.

/usr/share/groff/1.22.4/tmac/ps.tmac

Macros for use with grops; automatically loaded by troffrc

/usr/share/groff/1.22.4/tmac/pspic.tmac

Definition of PSPIC macro, automatically loaded by ps.tmac.

/usr/share/groff/1.22.4/tmac/psold.tmac

Macros to disable use of characters not present in older Post? Script printers (e.g., ?eth? or ?thorn?).

/tmp/gropsXXXXXX

Temporary file. See groff(1) for details on the location of temporary files.

SEE ALSO

afmtodit(1), groff(1), troff(1), pfbtops(1), groff_out(5), groff_font(5), groff_char(7), groff_tmac(5)

PostScript Language Document Structuring Conventions Specification ?http://partners.adobe.com/public/developer/en/ps/5001.DSC_Spec.pdf?