

NAME

gropdf – PDF driver for groff

SYNOPSIS

gropdf [**-dels**] [**-F** *dir*] [**-I** *dir*] [**-p** *paper-size*] [**-u** [*cmapfile*]] [**-y** *foundry*] [*file* ...]

gropdf **-v**

gropdf **--version**

DESCRIPTION

gropdf translates the output of GNU **troff** to PDF. Normally **gropdf** should be invoked by using the groff command with a **-Tpdf** option. If no files are given, **gropdf** reads the standard input. A filename of **-** also causes **gropdf** to read the standard input. PDF output is written to the standard output. When **gropdf** is run by **groff** options can be passed to **gropdf** using **groff**'s **-P** option.

See section “Font Installation” below for a guide how to install fonts for **gropdf**.

OPTIONS

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

- d** Include debug information as comments within the PDF. Also produces an uncompressed PDF.
- e** Forces **gropdf** to embed *all* fonts (even the 14 base PDF fonts).
- F** *dir* Prepend directory *dir*/devname to the search path for font, and device description files; *name* is the name of the device, usually **pdf**.
- I** *dir* This option may be used to add a directory to the search path for files named in **\X'pdf: pdfpic'** escape. The current directory is always searched first. This option may be specified more than once; the directories are then searched in the order specified.

No directory search is performed for files with an absolute file name.
- l** Orient the document in landscape format.
- p** *paper-size* Set physical dimension of output medium. This overrides the **papersize**, **paperlength**, and **paperwidth** commands in the *DESC* file; it accepts the same arguments as the **papersize** command. See **groff_font(5)** for details.
- s** Append a comment line to end of PDF showing statistics, i.e. number of pages in document. Ghostscript's **ps2pdf** complains about this line if it is included, but works anyway.
- u** [*cmapfile*] **Gropdf** normally includes a ToUnicode CMap with any font created using *text.enc* as the encoding file, this makes it easier to search for words which contain ligatures. You can include your own CMap by specifying a *cmapfile* or have no CMap at all by omitting the argument.
- v**
- version** Print the version number and exit.
- y** *foundry* Set the foundry to use for selecting fonts of the same name.

USAGE

The input to **gropdf** 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 multiple of 72 times the **sizescale**. The **pdf** 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. **gropdf** uses the same Type 1 Adobe PostScript fonts as the **grops** device driver. Although the PDF Standard allows the use of other font types (like TrueType) this implementation only accepts the Type 1 PostScript font. Fewer Type 1 fonts are supported natively in PDF documents than the standard 35 fonts supported by

grops and all PostScript printers, but all the fonts are available since any which aren't supported natively are automatically embedded in the PDF.

gropdf supports the concept of foundries, that is different versions of basically the same font. During install a *Foundry* file controls where fonts are found and builds **groff** fonts from the files it discovers on your system.

Each font description file must contain a command

internalname *psname*

which says that the PostScript name of the font is *psname*. Lines starting with # and blank lines are ignored. The code for each character given in the font file must correspond to the code in the default encoding for the font. 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 PostScript font, and the widths given in the font file must match the widths used in the PostScript font.

Note that **gropdf** is currently only able to display the first 256 glyphs in any font. This restriction will be lifted in a later version.

gropdf can automatically include the downloadable fonts necessary to print the document. Fonts may be in PFA or PFB format.

Any downloadable fonts which should, when required, be included by **gropdf** must be listed in the file */usr/share/groff/1.22.4/font/devpdf/download*; this should consist of lines of the form

foundry font filename

where *foundry* is the foundry name or blank for the default foundry. *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 must be separated by tabs (spaces are **not** allowed); *filename* is searched for using the same mechanism that is used for groff font metric files. The *download* file itself is also searched for using this mechanism; currently, only the first found file in the font path is used. Foundry names are usually a single character (such as 'U' for the URW Foundry) or blank for the default foundry. This default uses the same fonts as **ghostscript** uses when it embeds fonts in a PDF file.

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	<i>AvantGarde-BookOblique</i>
AB	AvantGarde-Demi
ABI	<i>AvantGarde-DemiOblique</i>
BMR	Bookman-Light
BMI	<i>Bookman-LightItalic</i>
BMB	Bookman-Demi
BMBI	<i>Bookman-DemiItalic</i>
CR	Courier
CI	<i>Courier-Oblique</i>
CB	Courier-Bold
CBI	<i>Courier-BoldOblique</i>
HR	Helvetica
HI	<i>Helvetica-Oblique</i>
HB	Helvetica-Bold
HBI	<i>Helvetica-BoldOblique</i>
HNR	Helvetica-Narrow
HNI	<i>Helvetica-Narrow-Oblique</i>
HNB	Helvetica-Narrow-Bold
HNBI	<i>Helvetica-Narrow-BoldOblique</i>
NR	NewCenturySchlbk-Roman

NI	<i>NewCenturySchlbk-Italic</i>
NB	NewCenturySchlbk-Bold
NBI	<i>NewCenturySchlbk-BoldItalic</i>
PR	Palatino-Roman
PI	<i>Palatino-Italic</i>
PB	Palatino-Bold
PBI	<i>Palatino-BoldItalic</i>
TR	Times-Roman
TI	<i>Times-Italic</i>
TB	Times-Bold
TBI	<i>Times-BoldItalic</i>

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. The lower case greek characters are automatically slanted (to match the SymbolSlanted font (SS) available to PostScript). Zapf Dingbats is available as **ZD**, the "hand pointing left" glyph ([lh]) is available since it has been defined using the `\X'pdf:xrev'` extension which reverses the direction of letters within words.

The default color for `\m` and `\M` is black.

gropdf understands some of the X commands produced using the `\X` escape sequences supported by **grops**. Specifically, the following is supported.

`\X'ps: invis'`

Suppress output.

`\X'ps: endinvis'`

Stop suppressing output.

`\X'ps: exec gsave currentpoint 2 copy translate n rotate neg exch neg exch translate'`

where *n* is the angle of rotation. This is to support the *align* command in **gpics**.

`\X'ps: exec grestore'`

Again used by **gpics** to restore after rotation.

`\X'ps: exec n setlinejoin'`

where *n* can be one of the following values.

0 = Miter join

1 = Round join

2 = Bevel join

`\X'ps: exec n setlinecap'`

where *n* can be one of the following values.

0 = Butt cap

1 = Round cap, and

2 = Projecting square cap

`\X'ps: ... pdfmark'`

All the *pdfmark* macros installed by using `-m pdfmark` or `-m mspdf` (see documentation in *pdfmark.pdf*). A subset of these macros are installed automatically when you use `-Tpdf` so you should not need to use `'-m pdfmark'` for using most of the PDF functionality.

gropdf also supports a subset of the commands introduced in *present.tmac*. Specifically it supports:-

PAUSE

BLOCKS

BLOCKE

Which allows you to create presentation type PDFs. Many of the other commands are already available in other macro packages.

These commands are implemented with **groff** X commands:-

\X'ps: exec % % % %PAUSE

The section before this is treated as a block and is introduced using the current **BLOCK** transition setting (see 'pdf: transition' below). This command can be introduced using the macro **.pdfpause**.

\X'ps: exec % % % %BEGINONCE

Any text following this command (up to % % % %ENDONCE) is shown only once, the next % % % %PAUSE will remove it. If producing a non presentation pdf, i.e. ignoring the pauses, see **GROPDF_NOSLIDE** below, this text is ignored.

\X'ps: exec % % % %ENDONCE

This terminates the block defined by % % % %BEGINONCE. This pair of commands is what implements the **.BLOCKS Once/.BLOCKE** commands in **present.tmac**.

The **mom** macro set already has integration with these extensions so you can build slides with **mom**.

If you use **present.tmac** with **gropdf** there is no need to run the program **presentps(1)** since the output will already be a presentation pdf.

All other **ps:** tags are silently ignored.

One **\X** special used by the DVI driver is also recognised:

\X'papersize=*paper-size*

where the *paper-size* parameter is the same as the **papersize** command. See **groff_font(5)** for details. This means that you can alter the page size at will within the PDF file being created by **gropdf**. If you do want to change the paper size, it must be done before you start creating the page.

In addition, **gropdf** supports its own suite of **pdf:** tags. The following tags are supported:

\X'pdf: pdfpic *file alignment width height line-length*

Place an image of the specified *width* containing the PDF drawing from file *file* of desired *width* and *height* (if *height* is missing or zero then it is scaled proportionally). If *alignment* is **-L** the drawing is left aligned. If it is **-C** or **-R** a *linelength* greater than the width of the drawing is required as well. If *width* is specified as zero then the width is scaled in proportion to the height.

\X'pdf: xrev'

This toggles a flag which reverses the direction of printing *letter by letter*, i.e., each separate letter is reversed, not the entire word. This is useful for reversing the direction of glyphs in the Dingbats font. To return to normal printing repeat the command again.

\X'pdf: markstart /ANN definition'

The macros which support PDF Bookmarks use this call internally to start the definition of bookmark hotspot (user will have called '**.pdfhref L**' with the text which will become the 'hot spot' region). Normally this is never used except from within the **pdfmark** macros.

\X'pdf: markend'

The macros which support PDF Bookmarks use this call internally to stop the definition of bookmark hotspot (user will have called '**.pdfhref L**' with the text which will become the 'hot spot' region). Normally this is never used except from within the **pdfmark** macros.

\X'pdf: marksuspend'

\X'pdf: markrestart'

If you are using page traps to produce headings, footings, etc., you need to use these in case a 'hot spot' crosses a page boundary, otherwise any text output by the heading or footing macro will be marked as part of the 'hot spot'. To stop this happening just place '**.pdfmarksuspend**' and '**.pdfmarkrestart**' at the start and end of the page trap macro, respectively. (These are just convenience macros which emit the **\X** code. These macros must only be used within page traps.)

\X'pdf: transition'feature mode duration dimension motion direction scale bool
where

feature can be either SLIDE or BLOCK. When it is SLIDE the transition is used when a new slide is introduced to the screen, if BLOCK then this transition is used for the individual blocks which make up the slide.

mode is the transition type between slides:-

Split - Two lines sweep across the screen, revealing the new page. The lines may be either horizontal or vertical and may move inward from the edges of the page or outward from the center, as specified by the *dimension* and *motion* entries, respectively.

Blinds - Multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines may be either horizontal or vertical, as specified by the *dimension*

entry. Horizontal lines move downward; vertical lines move to the right.

Box - A rectangular box sweeps inward from the edges of the page or outward from the center, as specified by the *motion* entry, revealing the new page.

Wipe - A single line sweeps across the screen from one edge to the other in the direction specified by the *direction* entry, revealing the new page.

Dissolve - The old page dissolves gradually to reveal the new one.

Glitter - Similar to Dissolve, except that the effect sweeps across the page in a wide band moving from one side of the screen to the other in the direction specified by the *direction* entry.

R - The new page simply replaces the old one with no special transition effect; the *direction* entry shall be ignored.

Fly - (PDF 1.5) Changes are flown out or in (as specified by *motion*), in the direction specified by *direction*, to or from a location that is offscreen except when *direction* is **None**.

Push - (PDF 1.5) The old page slides off the screen while the new page slides in, pushing the old page out in the direction specified by *direction*.

Cover - (PDF 1.5) The new page slides on to the screen in the direction specified by *direction*, covering the old page.

Uncover - (PDF 1.5) The old page slides off the screen in the direction specified by *direction*, uncovering the new page in the direction specified by *direction*.

Fade - (PDF 1.5) The new page gradually becomes visible through the old one.

duration is the length of the transition in seconds (default 1).

dimension (Optional; **Split** and **Blinds** transition styles only) The dimension in which the specified transition effect shall occur: **H** Horizontal, or **V** Vertical.

motion (Optional; **Split**, **Box** and **Fly** transition styles only) The direction of motion for the specified transition effect: **I** Inward from the edges of the page, or **O** Outward from the center of the page.

direction (Optional; **Wipe**, **Glitter**, **Fly**, **Cover**, **Uncover** and **Push** transition styles only) The direction in which the specified transition effect shall moves, expressed in degrees counterclockwise starting from a left-to-right direction. If the value is a number, it shall be one of: **0** = Left to right, **90** = Bottom to top (Wipe only), **180** = Right to left (Wipe only), **270** = Top to bottom, **315** = Top-left to bottom-right (Glitter only) The value can be **None**, which is relevant only for the **Fly** transition when the value of *scale* is not 1.0.

scale (Optional; PDF 1.5; **Fly** transition style only) The starting or ending scale at which the changes shall be drawn. If *motion* specifies an inward transition, the scale of the changes drawn shall progress from *scale* to 1.0 over the course of the transition. If *motion* specifies an outward transition, the scale of the changes drawn shall progress from 1.0 to *scale* over the course of the transition

bool (Optional; PDF 1.5; **Fly** transition style only) If **true**, the area that shall be flown in is rectangular and opaque.

This command can be used by calling the macro **.pdftransition** using the parameters described above. Any of the parameters may be replaced with a "." which signifies the parameter retains its previous value, also any trailing missing parameters are ignored.

Note: not all PDF Readers support any or all these transitions.

Importing graphics

gropdf only supports importing other PDF files as graphics. But that PDF file may contain any of the graphic formats supported by the PDF standard (such as JPEG, PNG, GIF, etc.). So any application which outputs PDF can be used as an embedded file in **gropdf**. The PDF file you wish to insert must be a single page and the drawing must just fit inside the media size of the PDF file. So, in **inkscape**(1) or **gimp**(1) (for example) make sure the canvas size just fits the image.

The PDF parser used in **gropdf** has not been rigorously tested with all possible applications which produce PDFs. If you find a single page PDF which fails to import properly, it is worth running it through the **pdftk**(1) program by issuing the command:

```
pdftk oldfile.pdf output newfile.pdf
```

You may find that *newfile.pdf* will now load successfully.

TrueType and other font formats

gropdf does not support any other fonts except Adobe Type 1 (PFA or PFB).

FONT INSTALLATION

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

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

The very first line in a PFA/PFB file contains this:

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

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

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

```
afmtodit Foo-Bar-Bold.afm map/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 'devpdf' 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 *devpdf/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) and belongs to foundry 'F' thus the following line should be added to *download*:

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

Use a tab character to separate the fields, and the 'foundry' field should be null for the default foundry.

ENVIRONMENT

GROFF_FONT_PATH

A list of directories in which to search for the *devname* directory in addition to the default ones. If, in the *download* file, the font file has been specified with a full path, no directories are searched. See **troff**(1) and **groff_font**(5) for more details.

GROPDF_NOSLIDE

If this is set true, **gropdf** will ignore all commands which produce a presentation pdf, and produce a normal pdf instead.

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/devpdf/DESC

Device description file.

/usr/share/groff/1.22.4/font/devpdf/F

Font description file for font *F*.

/usr/share/groff/1.22.4/font/devpdf/U-F

Font description file for font *F* (using foundry *U* rather than the default foundry).

/usr/share/groff/1.22.4/font/devpdf/download

List of downloadable fonts.

/usr/share/groff/1.22.4/font/devpdf/Foundry

A Perl script used during install to locate suitable fonts.

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

Encoding used for text fonts.

/usr/share/groff/1.22.4/tmac/pdf.tmac

Macros for use with **gropdf**; automatically loaded by **troffrc**.

SEE ALSO

afmtodit(1), **groff**(1), **troff**(1), **groff_font**(5), **groff_out**(5)