## NAME

pdfroff − generate Portable Document Format files using *groff*

## SYNOPSIS

**pdfroff** [*groff-option* . . .] [**−−no−toc−relocation**] [**−−no−kill−null−pages**[=*whence*]]
[**−−no−reference−dictionary**] [**−−reference−dictionary=***name*] [**−−report−progress**]
[**−−stylesheet=***name*] [**−−emit−ps**] [**−−no−pdf−output**] [**−−pdf−output=***name*]
[**−−keep−temporary−files**] [*file* . . .]

**pdfroff −h**
**pdfroff −−help**

**pdfroff −v** [*groff-option* . . .]
**pdfroff −−version** [*groff-option* . . .]

*groff-option* is any short option supported by groff(1) except for **−h**, **−T**, and **−v**; see section “Usage” below.

## DESCRIPTION

*pdfroff* is a wrapper program for the GNU text processing system, *groff*. It transparently handles the mechanics of multiple pass *groff* processing, when applied to suitably marked up *groff* source files, such that tables of contents and body text are formatted separately, and are subsequently combined in the correct order, for final publication as a single PDF document. A further optional “style sheet” capability is provided; this allows for the definition of content which is required to precede the table of contents, in the published document.

For each invocation of *pdfroff*, the ultimate *groff* output stream is post-processed by the Ghostscript gs(1) interpreter to produce a finished PDF document.

*pdfroff* makes no assumptions about, and imposes no restrictions on, the use of any *groff* macro packages which the user may choose to employ, in order to achieve a desired document format; however, it *does* include specific built in support for the *pdfmark* macro package, should the user choose to employ it. Specifically, if the *pdfhref* macro, defined in the *pdfmark.tmac* package, is used to define public reference marks, or dynamic links to such reference marks, then *pdfroff* performs as many preformatting *groff* passes as required, up to a maximum limit of *four*, in order to compile a document reference dictionary, to resolve references, and to expand the dynamically defined content of links.

## USAGE

The command line is parsed in accordance with normal GNU conventions, but with one exception—when specifying any short form option (i.e., a single character option introduced by a single hyphen), and if that option expects an argument, then it *must* be specified independently (i.e., it may *not* be appended to any group of other single character short form options).

Long form option names (i.e., those introduced by a double hyphen) may be abbreviated to their minimum length unambiguous initial substring.

Otherwise, *pdfroff* usage closely mirrors that of *groff* itself. Indeed, with the exception of the **−h**, **−v**, and **−T** *dev* short form options, and all long form options, which are parsed internally by *pdfroff*, all options and file name arguments specified on the command line are passed on to *groff*, to control the formatting of the PDF document. Consequently, *pdfroff* accepts all options and arguments, as specified in groff(1), which may also be considered as the definitive reference for all standard *pdfroff* options and argument usage.

## OPTIONS

*pdfroff* accepts all of the short form options (i.e., those introduced by a single hyphen), which are available with *groff* itself. In most cases, these are simply passed transparently to *groff*; the following, however, are handled specially by *pdfroff*.

**−h** Same as **−−help**; see below.

**−i** Process standard input, after all other specified input files. This is passed transparently to *groff*, but, if grouped with other options, it *must* be the first in the group. Hiding it within a group breaks standard input processing, in the multiple-pass *groff* processing context of *pdfroff*.

**−T** *dev* Of *groff*'s device selection options, only **−T pdf**, and **−T ps** are supported by *pdfroff*. **−T pdf** is silently remapped, to implicitly produce the effect of **−T ps**, and is otherwise ignored; *explicit* specification of **−T ps** retains its normal effect, but with the additional side effect of *implicitly* activating the effect of the **−−emit−ps** option, (see below). Attempting to specify any other device causes *pdfroff* to abort.

**−v** Same as **−−version**; see below.

See groff(1) for a description of all other short form options, which are transparently passed through *pdfroff* to *groff*.

All long form options (i.e., those introduced by a double hyphen) are interpreted locally by *pdfroff*; they are *not* passed on to *groff*, unless otherwise stated below.

**−h**, **−−help**
Causes *pdfroff* to display a summary of the its usage syntax, and supported options, and then exit.

**−T ps**, **−−emit−ps**
Suppresses the final output conversion step, causing *pdfroff* to emit PostScript output instead of PDF. This may be useful to capture intermediate PostScript output when using a specialised postprocessor, such as *gpresent* for example, in place of the default Ghostscript PDF writer.

Note that use of the **−Tps** option is implicitly enforced, when running *pdfroff*; if specified *explicitly*, it is interpreted as an alias for **−−emit−ps**.

**−−keep−temporary−files**
Suppresses the deletion of temporary files, which normally occurs after *pdfroff* has completed PDF document formatting; this may be useful when debugging formatting problems.

See section “Files” below for a description of the temporary files used by *pdfroff*.

**−−no−pdf−output**
May be used with the **−−reference−dictionary=***name* option (described below) to eliminate the overhead of PDF formatting when running *pdfroff* to create a reference dictionary for use in a different document.

**−−no−reference−dictionary**
May be used to eliminate the overhead of creating a reference dictionary, when it is known that the target PDF document contains no public references, created by the **pdfhref** macro.

**−−no−toc−relocation**
May be used to eliminate the extra *groff* processing pass, which is required to generate a table of contents, and relocate it to the start of the PDF document, when processing any document which lacks an automatically generated table of contents.

**−−no−kill−null−pages**[=*whence*]
While preparing for simulation of the manual collation step, which is traditionally required to relocate a *table of contents* to the start of a document, *pdfroff* accumulates a number of empty page descriptions into the intermediate PostScript output stream. During the final collation step, these empty pages are normally discarded from the finished document; this option forces *pdfroff* to leave them in place.

It may be noted that, in *pdfroff*'s default mode of operation, entirely blank pages are discarded *throughout* the finished document; the optional *whence* argument, which may take any *one* of the values, **toc**, **body**, or **all**, permits a finer degree of control over blank-page removal. The default behaviour, when **−−no−kill−null−pages** is specified *without* any argument, is equivalent to the effect of specifying **−−no−kill−null−pages=all**; this suppresses the removal of blank pages throughout the *entire* document. More often, it will be desired to leave blank pages in place within the body of the document, while removing them from the table of contents; this may be achieved by specifying **−−no−kill−null−pages=body**. Conversely, although perhaps less usefully, blank pages may be removed from the document body, while leaving them in place within the table of contents, by specifying **−−no−kill−null−pages=toc**.

**−−pdf−output=***name*

Specifies the name to be used for the resultant PDF document; if unspecified, the PDF output is written to standard output. A future version of *pdfroff* may use this option, to encode the document name in a generated reference dictionary.

**−−reference−dictionary=***name*

Specifies the name to be used for the generated reference dictionary file; if unspecified, the reference dictionary is created in a temporary file, which is deleted when *pdfroff* completes processing of the current document. This option *must* be specified, if it is desired to save the reference dictionary, for use in references placed in other PDF documents.

**−−report−progress**

Causes *pdfroff* to display an informational message on standard error, at the start of each *groff* processing pass.

**−−stylesheet=***name*

Specifies the name of an *input file*, to be used as a style sheet for formatting of content, which is to be placed *before* the table of contents, in the formatted PDF document.

**−v**, **−−version**

Causes *pdfroff* to display a version identification message. The entire command line is then passed transparently to *groff*, in a *one* pass operation *only*, in order to display the associated *groff* version information, before exiting.

## ENVIRONMENT

The following environment variables may be set, and exported, to modify the behaviour of *pdfroff*.

*PDFROFF_COLLATE*

Specifies the program to be used for collation of the finished PDF document.

This collation step may be required to move *tables of contents* to the start of the finished PDF document, when formatting with traditional macro packages, which print them at the end. However, users should not normally need to specify *PDFROFF_COLLATE*, (and indeed, are not encouraged to do so). If unspecified, *pdfroff* uses sed(1) by default, which normally suffices.

If *PDFROFF_COLLATE is* specified, then it must act as a filter, accepting a list of file name arguments, and write its output to the standard output stream, whence it is piped to the *PDFROFF_POSTPROCESSOR_COMMAND*, to produce the finished PDF output.

When specifying *PDFROFF_COLLATE*, it is normally necessary to also specify *PDFROFF_KILL_NULL_PAGES*.

*PDFROFF_COLLATE* is ignored, if *pdfroff* is invoked with the **−−no−kill−null−pages** option.

*PDFROFF_KILL_NULL_PAGES*

Specifies options to be passed to the *PDFROFF_COLLATE* program.

It should not normally be necessary to specify *PDFROFF_KILL_NULL_PAGES*. The internal default is a sed(1) script, which is intended to remove completely blank pages from the collated output stream, and which should be appropriate in most applications of *pdfroff*. However, if any alternative to sed(1) is specified for *PDFROFF_COLLATE*, then it is likely that a corresponding alternative specification for *PDFROFF_KILL_NULL_PAGES* is required.

As in the case of *PDFROFF_COLLATE*, *PDFROFF_KILL_NULL_PAGES* is ignored, if *pdfroff* is invoked with the **−−no−kill−null−pages** option.

*PDFROFF_POSTPROCESSOR_COMMAND*

Specifies the command to be used for the final document conversion from PostScript intermediate output to PDF. It must behave as a filter, writing its output to the standard output stream, and must accept an arbitrary number of *files* . . . arguments, with the special case of “**−**” representing the standard input stream.

If unspecified, *PDFROFF_POSTPROCESSOR_COMMAND* defaults to

```
gs -dBATCH -dQUIET -dNOPAUSE -dSAFER -sDEVICE=pdfwrite \
        -sOutputFile=-
```

*PDFWRITER*

If specified, this *must* represent one of the output device identifiers which is supported by the *PDFROFF_POSTPROCESSOR_COMMAND*; it overrides the default "*pdfwrite*" assignment within the "*−sDEVICE=pdfwrite*" specification, which itself appears as an argument within the definition of *PDFROFF_POSTPROCESSOR_COMMAND*.

*PDFWRITER* is ignored if the **−−emit−ps** option is specified, (or is implied, by explicit specification of the **−T ps** option); this has the effect of explicitly overriding *PDFWRITER*, such that it becomes equivalent to *PDFWRITER=ps2write*.

*GROFF_TMPDIR*

Identifies the directory in which *pdfroff* should create temporary files. If *GROFF_TMPDIR* is *not* specified, then the variables *TMPDIR*, *TMP* and *TEMP* are considered in turn as possible temporary file repositories. If none of these are set, then temporary files are created in the current directory.

*GROFF_GHOSTSCRIPT_INTERPRETER*

Specifies the program to be invoked when *pdfroff* converts *groff* PostScript output to PDF. If *PDFROFF_POSTPROCESSOR_COMMAND* is specified, then the command name it specifies is *implicitly* assigned to *GROFF_GHOSTSCRIPT_INTERPRETER*, overriding any explicit setting specified in the environment. If *GROFF_GHOSTSCRIPT_INTERPRETER* is not specified, then *pdfroff* searches the process *PATH*, looking for a program with any of the well known names for the Ghostscript interpreter; if no Ghostscript interpreter can be found, *pdfroff* aborts.

*GROFF_AWK_INTERPRETER*

Specifies the program to be invoked when *pdfroff* is extracting reference dictionary entries from a *groff* intermediate message stream. If *GROFF_AWK_INTERPRETER* is not specified, then *pdfroff* searches the process *PATH*, looking for any of the preferred programs, *gawk*, *mawk*, *nawk*, and *awk*, in that order; if none of these are found, *pdfroff* issues a warning message, and continues processing; however, in this case, no reference dictionary is created.

*OSTYPE*

Typically defined automatically by the operating system, *OSTYPE* is used on Microsoft Win32/MS-DOS platforms *only*, to infer the default *PATH_SEPARATOR* character, which is used when parsing the process *PATH* to search for external helper programs.

*PATH_SEPARATOR*

If set, *PATH_SEPARATOR* overrides the default separator character, (':' on POSIX/Unix systems, inferred from *OSTYPE* on Microsoft Win32/MS-DOS), which is used when parsing the process *PATH* to search for external helper programs.

*SHOW_PROGRESS*

If this is set to a non-empty value, then *pdfroff* always behaves as if the **−−report−progress** option is specified on the command line.

## FILES

Input and output files for *pdfroff* may be named according to any convention of the user's choice. Typically, input files may be named according to the choice of the principal normatting macro package, e.g., file.*ms* might be an input file for formatting using the *ms* macros (*s.tmac*); normally, the final output file should be named file.*pdf*.

Temporary files created by *pdfroff* are placed in the file system hierarchy, in or below the directory specified by environment variables (see section "Environment" above). If mktemp(1) is available, it is invoked to create a private subdirectory of the nominated temporary files directory, (with subdirectory name derived from the template *pdfroff−XXXXXXXXXX*); if this subdirectory is successfully created, the temporary files will be placed within it, otherwise they will be placed directly in the directory nominated in the environment.

All temporary files themselves are named according to the convention *pdf*$$.*, where *$$* is the standard shell variable representing the process identifier of the *pdfroff* process itself, and * represents any of the extensions used by *pdfroff* to identify the following temporary and intermediate files.

*pdf*$$.*tmp*
A scratch pad file, used to capture reference data emitted by *groff*, during the *reference dictionary* compilation phase.

*pdf*$$.*ref*
The *reference dictionary*, as compiled in the last but one pass of the *reference dictionary* compilation phase; (at the start of the first pass, this file is created empty; in successive passes, it contains the *reference dictionary* entries, as collected in the preceding pass).

If the **−−reference−dictionary**=*name* option is specified, this intermediate file becomes permanent, and is named *name*, rather than *pdf*$$.*ref*.

*pdf*$$.*cmp*
Used to collect *reference dictionary* entries during the active pass of the *reference dictionary* compilation phase. At the end of any pass, when the content of *pdf*$$.*cmp* compares as identical to *pdf*$$.*ref*, (or the corresponding file named by the **−−reference−dictionary**=*name* option), then *reference dictionary* compilation is terminated, and the *document reference map* is appended to this intermediate file, for inclusion in the final formatting passes.

*pdf*$$.*tc*
An intermediate *PostScript* file, in which “Table of Contents” entries are collected, to facilitate relocation before the body text, on ultimate output to the *Ghostscript* postprocessor.

*pdf*$$.*ps*
An intermediate *PostScript* file, in which the body text is collected prior to ultimate output to the *Ghostscript* postprocessor, in the proper sequence, *after pdf*$$.*tc*.

## AUTHORS

*pdfroff* was written by Keith Marshall ⟨keith.d.marshall@ntlworld.com⟩, who maintains it at his *groff-pdfmark* OSDN web-site ⟨https://osdn.net/users/keith/pf/groff-pdfmark/wiki/FrontPage⟩. The version distributed with *groff* is no longer maintained, and may be withdrawn in a future release.

## SEE ALSO

“*Groff: The GNU Implementation of troff*”, by Trent A. Fisher and Werner Lemberg, is the primary *groff* manual. You may browse it interactively, by running the “*info groff*” command.

Since *pdfroff* provides a superset of all of *groff*’s capabilities, the above manual, or its terser reference pages, groff(1), and groff(7), may also be considered definitive references to all *standard* capabilities of *pdfroff*, with this *pdfroff*(1) manual page providing the reference to *pdfroff*’s extended features.

While *pdfroff* imposes neither any restriction on, nor any requirement for, the use of any specific *groff* macro package, a number of supplied macro packages, and in particular those associated with the package *pdfmark.tmac*, are best suited for use with *pdfroff* as the preferred formatter. Detailed documentation on the use of these packages may be found, in PDF format, in the reference guide “*Portable Document Format Publishing with GNU Troff*”, written by Keith Marshall, which is nominally included in the installed documentation set, as */usr/local/share/doc/groff-pdfmark/pdfmark.pdf*. This file, together with its source, *pdfmark.ms*, is included within the *groff-pdfmark* distribution.