NAME

pdfroff − a macro for in-document control of pdfroff program features

DESCRIPTION

pdfroff is a wrapper program for the GNU text processing system, **groff**(1); its operation is supported by an accompanying macro, also named **pdfroff**, which provides capabilities for manipulation of a number of the **pdfroff**(1) program's feature control options, from within **groff**(7) document source.

The **pdfroff**(1) program is designed to facilitate the publication of PDF documents, and incorporates features which offer specific support for the use of special PDF mark-up attributes. For documentation of the **pdfroff** program, see **pdfroff**(1); the remainder of this manual page documents the use of the **pdfroff** macro.

USAGE

The **pdfroff**(1) program requests loading of the **pdfroff** macro, from its macro library file, when it runs each of its **groff**(1) processing passes; thus, provided the program, and its requisite macro files, have been correctly installed, no specific user action is necessary, to make the macro available for use. On the other hand, the **pdfroff** macro is not intended to be loaded, *unless* requested by the **pdfroff**(1) program; thus, testing for an available definition of the **pdfroff** macro is an effective method of confirming that a given document is being processed by the **pdfroff**(1) program.

In addition to supporting the majority of **groff**(1) options — which are simply passed on to **groff**(1) itself — the **pdfroff**(1) program also supports a number of extra options, each of which is specific to the control of its own operation. Although any of these **pdfroff**(1) specific options may be specified on the command line, it may often be convenient to specify some of them from within the document source; this may be achieved by using the **pdfroff** macro:

.pdfroff option <*equivalent-variable-name*>[=<*option-setting*>]

As currently implemented, this macro supports control of only a few of the **pdfroff**(1) optional features. Those which *are* supported are limited to:

.pdfroff option preserve_blank_pages[=*all*|*toc*|*body*]
.pdfroff option toc_relocation[=*auto*|*disabled*|*enabled*]
.pdfroff option toc_file=<*file-name*>

The first of these is a direct analogue of the **pdfroff**(1)

−−no−kill−null−pages[=*all*|*toc*|*body*]

command line option, which may be found useful to modify the default behaviour of removing *all* entirely blank pages from the document output stream; when preparing the document for printing on a duplex-capable device, it may be desireable to retain such pages, particularly within the document's *body* context.

The other forms of **pdfroff** macro usage relate to table of contents collation; the form:

.pdfroff option toc_relocation=auto

is a reflection of the default **pdfroff**(1) initial state, with respect to the pending operation of the traditional "print-at-end and collate manually" technique. In early (now obsolete) versions of **pdfroff**(1) this initial default state was equivalent to:

.pdfroff option toc_relocation=enabled

and it needed to be disabled, by invoking **pdfroff**(1) with its **−−no−toc−relocation** command line option, when formatting documents which were not specifically engineered to make use of this collation technique. More recent (current) versions of **pdfroff**(1) now start in an initial default state which is equivalent to:

.pdfroff option toc_relocation=disabled

which is analogous to the state established by the **−−no−toc−relocation** option, (thus making this option effectively redundant). The **enabled** state *must* now be requested, *explicitly*, for any document wishing to avail itself of the **pdfroff**(1) emulation of the "print-at-end and collate manually" technique;

the designation of the initial **disabled** state as **auto** is indicative that the enabling request:

.pdfroff option toc_relocation=enabled

is expected to originate in some macro, (possibly itself originating in a standardized package), which is associated with this collation technique.

Also associated with collation of tables of contents, but intended to facilitate the implementation of a — potentially more effective — alternative to the default **pdfroff**(1) "print-at-end" collation technique, a request in the form:

.pdfroff option toc_file=<*file-name*>

instructs **pdfroff**(1) to filter all single-line records of the form:

.toc anything . . .

from its standard error data stream, and to write copies of these collected records into the named file; this may then be included within the **groff**(7) document input data stream, (using the *.so* request, for example), whence its content may be interpreted as a sequence of macro calls, to lay out a table of contents, at the point where the file is included.

The content of the named *toc_file* may be written by inserting **.tm** requests, at appropriate points within the document source, in the form:

.tm .toc *record content . . .*

In this case, the record *must* begin with the **.toc** request keyword, but the remainder of the "*record content . . .*" is at the discretion of the document author, who must also assume responsibility for defining the **.toc** macro, which will subsequently interpret that record content, when it is read back from the document source input stream.

Alternatively, the document author may elect to adopt a macro package, such as a derivative of the **groff_toc**(7) framework, both to write the content of the named *toc_file*, and to interpret that content, when it is read back, (although it may still be necessary to write additional macros, to facilitate the interpretation); in this case, the form of the content will be dictated by the chosen macro implementation.

CONTROL REGISTERS

While processing any **groff**(7) input data stream, **pdfroff**(1) initializes a register named **PHASE**, and passes it to each invocation of **groff**(1) which it initiates; the particular value passed to each invocation depends on the particular phase of **pdfroff**(1) processing in which the invocation occurs:

0. **pdfroff**(1) is analysing the document layout, compiling a reference dictionary, and optionally, collecting table of contents records; the **pdfroff** macro will produce output *only* when the **PHASE** register is in this state, (or is undefined).
1. **pdfroff**(1) is compiling a table of contents section, emulating the "print-at-end with manual collation" technique, for eventual inclusion in the finished document.
2. **pdfroff**(1) is formatting the body of the document, which will ultimately be combined with any front-matter, which may have been compiled as directed by the **--stylesheet** option, and with any separately compiled table of contents, to produce the finished document.

When running **pdfroff**(1), the definition of the **pdfroff** macro is accompanied by *two* mnemonic register definitions, namely **PDF-TOC-ONLY**, and **PDF-BODY-ONLY**. These may used in any context where a mnemonic representation of the **PHASE** value for phases 1, and 2 respectively, is desired; each of these, together with the **PHASE** register itself, should be considered to be *constant valued*.

FILES

/usr/local/share/groff/site-tmac/pdfroff.tmac

This implements the **pdfroff** macro, and defines the mnemonic registers, **PDF-TOC-ONLY**, and **PDF-BODY-ONLY**; it is loaded *automatically*, when **pdfroff**(1) invokes **groff**(1).

CAVEATS AND BUGS

The **pdfroff** macro is an experimental feature, which was introduced in the *20030406.1* release of *groff-pdfmark*; it is *not* supported in any version of **pdfroff**(1) from any earlier release, (including the obsolete version which continues to be distributed by the GNU Troff project).

Only a few of the **pdfroff**(1) command line options may be controlled using the **pdfroff** macro; the subset for which this is possible is limited to those described in the **USAGE** section, above.

There is (currently) no **pdfroff**(1) command line equivalent for the macro assignment:

.pdfroff option toc_ file=<file-name>

consequently, if such an assignment is required, it may *only* be assigned by use of the **pdfroff** macro.

Only minimal validation of **pdfroff** macro arguments is performed. A warning diagnostic will be issued, if the first argument is not a valid keyword, (with only **option** being considered as valid, at present). Beyond this, any arbitrary text will be accepted for the <*equivalent-variable-name*> and its associated <*option-setting*> arguments; however, the assignment will be silently ignored, if the <*equivalent-variable-name*> does not match one of those documented in the **USAGE** section, while the behaviour will be undefined if the <*equivalent-variable-name*> does match one of those documented, but the associated <*option-setting*> does not match any of the expected values.

Although it is stipulated above, in the **CONTROL REGISTERS** section, that each of the **PHASE**, **PDF-TOC-ONLY**, and **PDF-BODY-ONLY** registers should be considered to be *constant valued*, the **groff**(7) language provides no mechanism to enforce this; consequently, users are cautioned that they should avoid changing the value of any of these registers, as the effect of doing so may result in undefined behaviour.

EXAMPLES

To verify that a particular document is being processed by **pdfroff**(1):

```
.if !d pdfroff \
.   ab "Please use pdfroff to format this document."
```

To maintain correct recto/verso pagination, by insertion of, and non-removal of entirely blank pages, when printing hard-copy using both sides of the paper, (i.e. duplex printing), follow the preceding verification check with:

```
.if duplex .pdfroff option preserve_blank_pages=all
```

and specify *−duplex* as a command line option, when invoking **pdfroff**(1).

AUTHORS

The **pdfroff** macros are provided by the *groff-pdfmark* package, which was written by Keith Marshall <author@address.hidden>; it is independently maintained at Keith's *groff-pdfmark* project hosting web-site <https://savannah.nongnu.org/projects/groff-pdfmark/>, whence the latest version may *always* be obtained.

SEE ALSO

groff(1), **pdfroff**(1), **groff**(7), **groff_toc**(7)

More comprehensive documentation, on the use of **pdfroff**, and the *groff-pdfmark* macro suite may be found, in PDF format, in the reference guide "*Portable Document Format Publishing with GNU Troff*", which has also been written by Keith Marshall; the most recently published version of this guide may be read online, by following the appropriate document reference link on the *groff-pdfmark* project hosting web-site <https://savannah.nongnu.org/projects/groff-pdfmark/>, whence a copy may also be downloaded.