NAME

groff_pdfnote – insert virtual "sticky notes" in groff PDF documents

DESCRIPTION

This manual page serves as a formal reference for the use of the **pdfnote** macro, as implemented for the **groff**(7) document processing language. This macro is provided by the **groff_pdfmark**(7) package; it offers a mechanism whereby an author may insert *editorial annotations*, in the form of virtual "sticky notes", into any PDF document which is created by **groff**(1).

USAGE

The **pdfnote** macro is made available for use within any **groff**(7) input data stream into which the **groff_pdfmark**(7) macro package is incorporated, either by inclusion from the **groff**(1), or the **pdfroff**(1) command line:

groff –m pdfmark [*option* . . .] [*file* . . .]
pdfroff –m pdfmark [*option* . . .] [*file* . . .]

or by inclusion within the **groff**(7) source data itself:

.mso pdfmark.tmac

Once the **groff_pdfmark**(7) macro package has been loaded, virtual "sticky note" annotations may be inserted into any **groff**(1) PDF output data stream — or **pdfroff**(1) output stream — by invocation of a macro call of the form:

.pdfnote [[**–T** "*title text* . . ."] [\[**#** *comment* . . .]]
[**–PD** <*paragraph-spacing-line-count*>] [\[**#** *comment* . . .]]
[**-C** <*red-value*> <*green-value*> <*blue-value*>] [\[**#** *comment* . . .]]
[**–O**] [**-I** <*icon-style-name*>] [**––**] [\[**#** *comment* . . .]]
[*first of multiple lines of annotation text* . . . \[**#** *comment* . . .]]
[*additional non-terminal line(s) of annotation text* . . . \[**#** *comment* . . .]]
[. . .]
] *last (or only) line of annotation text* . . .

with options:

–T "*title text* . . ."
Set "*title text* . . ." within the title bar of the PDF annotation, when it is presented in its *open* state. Note that "*title text* . . ." *must* be specified as a single argument to the **pdfnote** macro, and thus, *must* be quoted as such, (by enclosing it between ASCII double quotation mark tokens), *if* it comprises multiple lexical words, separated by white space.

–O Register a preference that the associated **pdfnote** should initially be presented in its *open* state, when the containing PDF document is opened. (Note that there is no guarantee that this preference will actually be honoured, by whichever PDF viewer has been chosen to open the document).

–I <*icon-style*>
Specify the style of icon, which is to be associated with a particular **pdfnote**; this should be identified by the <*icon-style*> argument, which *must* be specified as any *one* of the icon style names supported by the PDF viewer which is in use, (noting that *all* PDF viewers are *required* to support each of the styles: *Note*, *Comment*, *Help*, *Insert*, *Key*, *NewParagraph*, and *Paragraph*, with *Note* representing the default icon style).

–C <*red-value*> <*green-value*> <*blue-value*>
Specifies the desired background colour for the virtual "sticky note" panel, and its associated icon. The desired colour is specified as a triplet of numerical values, in the RGB colour space, with each of the <*red-value*>, <*green-value*>, and <*blue-value*> arguments being expressed as a *positive* decimal value in the range 0.0 to 1.0; (subject to the limitation that the capability for interpretation of such RGB colour specifications may not be fully supported by *all* PDF viewer applications).

–PD <*paragraph-spacing*>
Specifies the number of blank lines, (a *positive whole number,* with the default being *zero*), which should be used to separate paragraphs, when a new paragraph mark, (see section on **CONTROL SETTINGS**), is inserted into the body of a **pdfnote**.

`--` Suppress interpretation of any of the further macro arguments, which have been passed in the current **.pdfnote** call, as a potential macro option; this can be useful when the argument which follows has a form resembling an option, but is intended to be be included within the body text of the **pdfnote**.

Notice that each individual **pdfnote** *must* be represented, within the **groff**(7) document source data stream, as a *single* macro call. Since it will often be inconvenient to express the *entire* content of many **pdfnote**s, on just one *physical* input line for each, a single *logical* input line may be constructed, by ending each *physical* input line, from the initiating **.pdfmark** call line, until, but *excluding*, the last line of the **pdfnote** body, with an escaped newline — or with **groff**(7)'s "`\#`" escape sequence, and an optional trailing comment.

CONTROL SETTINGS

In addition to the option arguments, as described in the preceding **USAGE** section, the **pdfnote** macro interprets the following control registers, to determine the placement, and the layout, of each **pdfnote** annotation, which is inserted into the PDF document output stream:

PDFNOTE.OFFSET

This is defined as a **groff**(7) string variable; its value represents an expression, which, when evaluated, computes the offset, in device units, relative to the last text output position on the current page, of the left-hand boundary of the page region in which the annotation should be placed. Defined, by default, as:

```
.ds PDFNOTE.OFFSET "\\n[.k]+\\n[.o]+\\n[.in]\"
```

which causes **pdfnote** annotations to be embedded, within the running text, *immediately* following the glyph which was, most recently, inserted into the PDF text output stream.

This default placement of **pdfnote** annotations will tend to occlude parts of the running text, on the PDF output page. It can be challenging to circumvent such occlusion; thus, users may prefer to redefine **PDFNOTE.OFFSET**, to relocate **pdfnote** annotations, for example, to the left-hand page margin:

```
.ds PDFNOTE.OFFSET "\\n[.o]-\\n[PDFNOTE.WIDTH]-1m\"
```

thus computing the placement for the left-hand side of the **pdfnote** region, within the left-hand page margin, (as defined by the "**.po**" request), accounting for the nominal width of the **pdfnote** region itself, and allowing for 1em of space between the right-hand side of the **pdfnote** region and the leftmost extent of the running text; it *assumes* that the margin width is sufficient to accommodate this, *but does not verify* this assumption.

PDFNOTE.LEADING

Complementary to the **PDFNOTE.OFFSET** control, **PDFNOTE.LEADING** establishes the placement for the topmost edge of the **pdfnote** display region; it represents, in device units relative to the baseline of the line of running text which was most recently written to the PDF output stream, the displacement of the top of the **pdfnote** display region, and is defined, by default, as:

```
.nr PDFNOTE.LEADING 0.3v
```

which sets the top of the **pdfnote** display region to be 30% of the current line height *below* the baseline of the preceding line of running text.

Notice that, whereas **PDFNOTE.OFFSET** is defined as a *string*, which will ultimately be interpreted as a numeric expression, **PDFNOTE.LEADING** is defined, from the outset, as a simple numeric value; this distinction should be borne in mind, if it is desired to redefine either **PDFNOTE.OFFSET** or **PDFNOTE.LEADING**.

PDFNOTE.WIDTH
PDFNOTE.HEIGHT

With default definitions of:

```
.nr PDFNOTE.WIDTH  0.8c
.nr PDFNOTE.HEIGHT 0.9c
```

this pair of numeric registers defines the width, and the height respectively, of each **pdfnote** display region; when they are evaluated in conjunction with the **PDFNOTE.OFFSET** and

PDFNOTE.LEADING registers, the placement and extent of each **pdfnote** display region is completely specified.

PDFNOTE.QUOTED

Since the entire content of any **pdfnote** *must* be passed as a list of arguments, in a **pdfnote** macro call, and given that, when any ASCII double quote character is included in such an argument list, it is commonly preceded by white space, which causes it to be interpreted as an argument grouping mark; in this context, it can be challenging to specify such quoting characters with the intent that they should appear within the body text of any **pdfnote**.

While it may be possible to specify literal ASCII quotation marks, by placing *two* of them consecutively, for each individual literal mark, within a quoted argument group, a possibly more intuitive alternative is offered by the **PDFNOTE.QUOTED** string. Defined as a template, (which should be considered to be represented as a *string constant*), this may be interpolated, within the body text region of any **pdfnote** macro argument list, in the form:

```
\*[PDFNOTE.QUOTED text to be quoted ...]
```

which results in insertion of "`text to be quoted ...`", enclosed between literal ASCII double quotation marks, into the text of the resultant **pdfnote**.

PDFNOTE.NEWLINE

Even more so, than the inclusion of literal ASCII double quotation marks within the text of any **pdfnote**, it can be extremely challenging to insert hard line breaks. This challenge may be overcome, by interpolation of:

```
\*[PDFNOTE.NEWLINE]
```

at appropriate points within the text region of the arguments passed to the **pdfnote** macro.

As in the case of **PDFNOTE.QUOTED**, the definition of **PDFNOTE.NEWLINE** is also formulated as a **groff**(7) string, which should be considered to represent a *string constant*.

PDFNOTE.PILCROW

Named to reflect its association with the traditional typographer's end-of-paragraph mark, this is a specialization of the **PDFNOTE.NEWLINE** feature. Interpolated within **pdfnote** macro arguments, in similar fashion:

```
\*[PDFNOTE.PILCROW]
```

its effect differs from that of **PDFNOTE.NEWLINE** only insofar as it repeats the effect of **PDFNOTE.NEWLINE**, to introduce the paragraph break, and then as many additional times as have been specified for the <*paragraph-spacing*> argument, in the most recent **pdfnote** macro call, in which the "**–PD**" option was specified.

Once again, the definition of **PDFNOTE.PILCROW** should be considered to represent a *string constant*.

FILES

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

This implements the core functionality of the entire *groff-pdfmark* macro suite, including the implementation of, and supporting infrastructure for, the **pdfnote** macro.

AUTHORS

The **pdfnote** macro is provided by the *groff-pdfmark* package, which was written by Keith Marshall <keith@address.hidden>; formerly distributed with *GNU roff*, it is now independently maintained at, and distributed from 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_pdfmark**(7)

More comprehensive documentation, on the use of 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.