Cooking up Enhanced PDF with pdfmark Recipes

GLOBAL GRAPHICS®
software

By
Lynn Mead
PDF Product Expert
Global Graphics Software

The information in this publication is provided for information purposes only and is subject to change without notice. The information is provided on an "AS IS" basis without any representations or any kind of warranty (whether express, implied or statutory), including the implied warranties of merchantability quality or fitness for a particular purpose.)

Global Graphics Software will not be liable for any damages, including indirect or consequential damages, or any damages arising from use or loss of use, data or profits, whether in contract, negligence or other tortious action, arising from or in connection with the use of this information.

The software described in this book is furnished under license and may only be used or copied in accordance with the terms of that License

Contents

Introduction

This book in intended as an introduction to the use of pdfmarks to add enhancements to PDF files during the PDF generation process. Not every aspect of using pdfmarks is covered here. I have tried to include what I think are the easiest and most common uses of pdfmarks.

The information presented in this book is based on the Adobe® PDFmark Reference Manual (technical note # 5150), and the PDF Reference version 1.4. My goal in writing this “Recipe Book” is to make this information easier for the “non-programmer” to understand and use. Hopefully the recipes presented here will inspire you to create pdfmarks tailored to your own needs.

Lynn Mead

Global Graphics Software (http://www.globalgraphics.com) is a leading developer and supplier of sophisticated high performance RIPs, PDF document, workflow and color solutions mostly to a customer base of Original Equipment Manufacturers, system integrators and resellers. These partners include the world's leading vendors of digital pre-press systems, large-format color printers, color proofing systems, copiers and printers for the corporate and SOHO markets. Jaws PDF creation tools have been developed since the format's inception in 1994.

Overview

What are pdfmarks?

pdfmarks are an extension of PostScript® code used to represent PDF features.

PDF features defined by pdfmarks are automatically generated when the PostScript code is converted into a PDF.

Only PostScript interpreters designed to create PDF files, like Jaws PDF Creator^{TM} or Acrobat® Distiller^{TM}, can interpret pdfmarks.

What can you do with pdfmarks?

pdfmarks enable you to:

- **ADD** annotations, links, form fields, bookmarks, articles, named destinations, page transitions
- **SET** page cropping, to open to a specific page, to open with bookmarks or thumbnails displayed
- **DEFINE** page labels, document information fields

In other words many of the enhancements that make PDFs dynamic and interactive can be implemented with pdfmarks.

Why use pdfmarks?

pdfmarks allow you to save time through Automation!

- Automatically generate PDF features instead of adding them manually.
- No re-creating features manually each time you update a document.
- Working with document collections can be easier and quicker with pdfmarks.

Understanding pdfmarks

The pdfmarks have three parts:

1. Every pdfmarks starts with the mark **object** - this is the **[** (left bracket) character.

2. **Arguments** - these describe what the pdfmark's features are. These are the ingredients of the pdfmark recipe.

3. A **name** - this specifies the *kind* of pdfmark.

The following is an example of a note pdfmark showing the three parts:

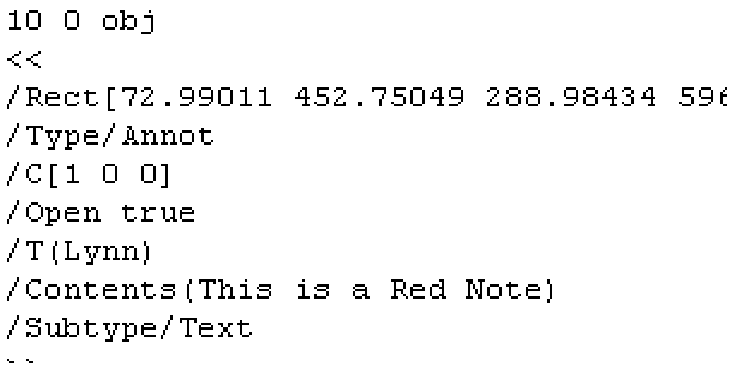
```
mark object-----    [

                    /Rect [ 0 0 216 144 ]
                    /Open true
                    /Title (Lynn)
arguments------     /Contents (This is a Red Note)
                    /Color [1 0 0 ]
                    /Subtype /Text

name------------    /ANN pdfmark
```

Note: Most pdfmark arguments resemble PDF dictionary keys. This is an important fact to remember. If you are unsure how to construct a specific argument sometimes you can open a pdf that contains a similar feature and determine from the dictionary entry (reverse engineer) how the pdfmark entry should look.

Following is an example of a dictionary entry for a note similar to the one described in the pdfmark above. You can see the similarities.



```
10 0 obj
<<
/Rect[72.99011 452.75049 288.98434 59
/Type/Annot
/C[1 0 0]
/Open true
/T(Lynn)
/Contents(This is a Red Note)
/Subtype/Text
>>
```

Keys to Success

The “mark object” ([) and “name” (pdfmark type) elements of the pdfmark are straightforward and easy to implement. “Arguments” however can appear confusing to a non-programmer. The following section is intended to “decode” arguments and provide you with the understanding needed to define features represented by arguments. Once you gain an understanding of arguments you can edit the pdfmark recipes in this book and customize them for your own needs.

Key-Value pairs

Arguments are expressed as Key-Value pairs. Each type of pdfmark uses a set of required and optional key-value pairs. Some key-value pairs are specific to a type of pdfmark, some are used more globally.

Keys start with a slash /, followed by the name of the key (first character capitalized). Some examples of keys are:

/Rect
/Color
/Page

Values can be expressed in several different formats.

string: A string of characters within parentheses that can be letters or numbers, for example, the content of a note is defined as a string (This represents the contents of a note). When using parentheses themselves, line-end characters, or backslashes inside strings, to be recognized as part of the string they must be preceded by a backslash. **Example:** (strings containing text in parentheses \(like this\))

Array: An array’s elements may be any combination of numbers, strings, dictionaries, or any other objects, including other arrays. An array is written as a sequence of objects enclosed in square brackets. **For example:** color is defined as an array of three numbers [1 0 0] and coordinates as an array of four numbers [0 0 144 72].

integer: whole numbers. You would define the page a link would jump to by using an integer. **Example:** /Page 2.

Boolean: A conditional item. May be a switch, such as on or off, expressed as true for on and false for off. You would use a boolean value to define whether a note was to display open or closed. **Example:** */Open true* displays the note open, */Open false* displays the note closed.

name: An identifier for a function or a parameter, always starting with a slash “/”. Names can be up to 127 characters long (including the slash). Names must not contain %()<>[]{}/ or # characters or special characters. **Example:** /Next

Specifying page numbers

All pages in a PDF document are numbered sequentially; the first page in a document is page 1.

When referring to pages in a pdfmark recipe, all page numbers must be specified using this sequence number, not the page number as it appears on the printed page.

	Document page #	Refer to PDF page #
Cover	none	1
Front Matter	i, ii, iii	2, 3, 4
Chapter 1	1 through 10	5 through 14

Example of an argument specifying a page number:
/Page 2

Defining Colors

1. Colors are defined by an array of three numbers placed between brackets that represent RGB values and look something like this [1 0 .65]

2. These numbers must be 0, 1 or any decimal in between and represent percentages of Red, Green and Blue.
   0 = 0% color 1 = 100% color .5 = 50% color

3. Percentages between 0 and 100% are defined by decimals. There is no required number of digits to the right of the decimal point. 1, .6, ,.25, .824 are all valid. Following are some examples of color definitions.

Color		Defined as	Red	Green	Blue
Black	⬛	[0 0 0]	0%	0%	0%
White	⬜	[1 1 1]	100%	100%	100%
Red	🟥	[1 0 0]	100%	0%	0%
Green	🟩	[0 1 0]	0%	100%	0%
Blue	🟦	[0 0 1]	0%	0%	100%
Cyan	🟦	[0 1 1]	0%	100%	100%
Magenta	🟪	[1 0 1]	100%	0%	100%
Yellow	🟨	[1 1 0]	100%	100%	0%
50% Grey	◼	[.50 .50 .50]	50%	50%	50%
Brown	🟫	[.64 .16 .16]	64%	16%	16%
Mint	🟩	[.2 .9 .45]	20%	90%	45%
Orange	🟧	[1 .5 .1]	100%	50%	10%
Pink	🟥	[1 .5 .5]	100%	50%	50%

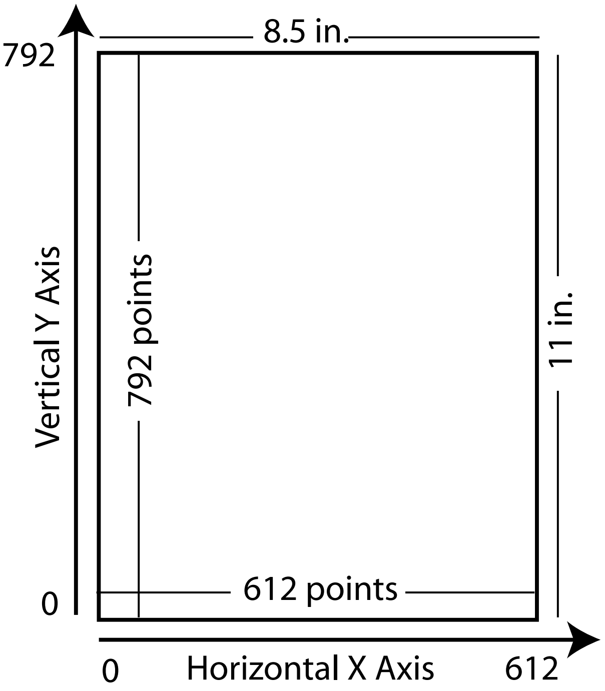
Example of an argument that defines color:
/Color [.3 .6 .734]

Defining Coordinates

The size and position of certain elements defined by pdfmarks are based on the coordinate system used in PostScript.

1. The origin point is the lower left-hand corner of the page. This is point 0,0.
2. The positive X-axis extends horizontally to the right of the origin point.
3. The positive Y-axis extends vertically upward from the origin point.
4. The unit of measure is a point

   1 point = 1/72 inch
   72 points = 1 inch
   8 1/2 x 11 inches = 612 x 792 points



A rectangle is described by an array of four numbers.

[X_{LL} Y_{LL} X_{UR} Y_{UR}]

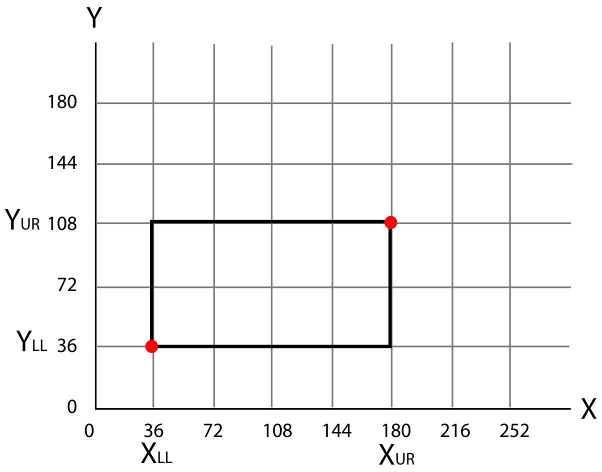
X_{LL} = lower left corner on the X axis

Y_{LL} = lower left corner on the Y axis

X_{UR} = upper right corner on the X axis

Y_{UR} = upper right corner on the Y axis

[X_{LL} Y_{LL} X_{UR} Y_{UR}] = [36 36 180 108]



LL=lower left corner UR=upper right corner

A rectangle that is 1″ tall and 2″ wide placed 1/2″ from the bottom and left page edge would be defined as [36 36 180 108]

Example of an argument defining location and size:
/Rect [108 108 216 180]

In some cases coordinates may be affected by the method of inserting the pdfmark.

1. When using FrameMaker you can enter code to make use of the PostScript Text Frame’s size and position. See “Using PostScript text frames in FrameMaker” on page 39 for more information.
2. When using Word you can use field code variables to make use of the paragraph or cell size and location. See “Using Field Codes in MS Word” on page 40 for more information.
3. When embedding pdfmarks in EPS files the size and location of the EPS is used. See “Placing pdfmarks inside an EPS file” on page 41 for more information.

Setting Views

To define how a page will display on the screen you use the /View key. The /View key can be used when defining links, named destinations, open options, or bookmarks. The following table describes the different values available, what they mean, and demonstrates how to enter them in your pdfmark.

Key	**Value**		**Explanation**	**Example**
	Name	**Parameters**		
/View	/Fit	---	Fit the page to the window.	/View [/Fit]
/View	/FitB	---	Fit the bounding box of the page contents to the window.	/View [/FitB]
/View	/FitH	top	Fit the width of the page to the window. *top* specifies the distance from the page origin to the top of the window. If top value is -32768 the top value is calculated automatically.	/View [/FitH -32768] /View [/FitH 5]
/View	/FitBH	top	Fit the width of the bounding box of the page contents to the window. *top* specifies the distance from the page origin to the top of the window.	/View [/FitBH 5]
/View	/FitR	x1 y1 x2 y2	Fit the rectangle specified by the parameters to the window.	/View [/FitR 30 648 209 761]
/View	/FitV	left	Fit the height of the page to the window. *left* specifies the distance in from the page origin to the left edge of the window.	/View [/FitV -5]
/View	/FitBV	left	Fit the height of the bounding box of the page contents to the window. *left* specifies the distance from the page origin to the left edge of the window.	/View / [/FitBV 18]
/View	/XYZ	left top	*left* and *top* specify the distance from the origin of the page to the top-left corner of the window. *zoom* specifies the zoom factor, with 1 being 100% magnification. specifying a view destination of /View [/XYZ null null null] goes to the specified page and retain the same horizontal and vertical offset and zoom as the current page. A zoom of 0 has the same meaning as a zoom of *NULL*.	/View [/XYZ 5 802 1.5] /View [/XYZ null null 0]

Assigning Actions

Following are some of the actions you can apply with pdfmarks. Not covered are Movies, Sounds, JavaScript and Submitting Form Data as these are more advanced topics. See "References" on page 45 for information on these actions.

Actions	Uses	Description	Example	Can be used with
Go To View	/Page /View /Dest	Links to another page and a defined or default view magnification Requires both /Page and /View keys or the /Dest key	[/Rect [65 488 180 510] /Page 2 /View [/Fit] /Subtype /Link /ANN pdfmark	Links Bookmarks
	/Action /GoToR	Links to another PDF file and a defined or default view magnification Requires the and /File key and /Page and /View keys or the /Dest key	[/Rect [65 488 180 510] /Action /GoToR /File (newdocument.pdf) /Page 2 /View [/XTYZ null null null] /Subtype /Link /ANN pdfmark	Links Bookmarks
Open File	/Action /Launch	Opens non-PDF file or an application. Requires /File key	[/Rect [65 488 180 510] /Action /Launch /File (word.doc) /Subtype /Link /ANN pdfmark	Links Bookmarks Form Fields
Execute Menu Items	/Action <</Subtype / Named /N /MenuItem >>	Executes a menu item	[/Rect [65 488 180 510] /Action << /Subtype /Named / N /Print >> /Subtype /Link /ANN pdfmark	Links Bookmarks Form Fields
World Wide Web Link	/Action << /Subtype /URI /URI (url of the file http://www....) >>	Links to a document using Universal Resource Locator	[/Rect [65 488 180 510] / Action << /Subtype /URI /URI (http://www.jawspdf.com/ pdfs/pdf_editor.pdf) >> /Subtype /Link /ANN pdfmark	Links Bookmarks Form Fields
Read Article	/Action /Article	Begins the process of following a selected article thread Requires /Dest key and if in another document /File key	[/Dest (My Story) / Title (My Story) /Action /Article /File (StoryCollection.pdf) /Subtype /Link /ANN pdfmark	Links Bookmarks Form Fields

Actions	Uses	Description	Example	Can be used with
Show/Hide Field	/Action << /Subtype /Hide /T (field name) /H true >>	Shows/Hides the specified fields	[/Rect [65 488 180 510] // Action << /Subtype /Hide /T (Help) /H true >> /Subtype /Link /ANN pdfmark	Form Fields
Import Form Data	/Action << /Subtype /ImportData /F (filename.fdf) >>	Imports an FDF file to populate a PDF form.	[/Rect [65 488 180 510] /Action << /Subtype /ImportData /F (myfile.fdf) >> /Subtype /Link /ANN pdfmark	Links Bookmarks Form Fields
Reset Form	/Action << /Subtype /ResetForm >>	Resets the values of all form fields to their defauls or to empty if no default is set.	[/Rect [65 488 180 510] /Border [0 0 1] /Action << /Subtype / ResetForm >> /Subtype /Link /ANN pdfmark	Links Bookmarks Form Fields

The Recipes

Writing and editing pdfmarks

Following are some important things you need to know about writing and editing pdfmarks:

1. When writing or editing pdfmarks remember to save in text only format. You can add a .ps or .eps extension as necessary and still save as plain text.

2. Syntax is important in PostScript and pdfmarks. Make sure that words are capitalized correctly or are preceded or surrounded by the proper characters. If your pdfmark is not working, check to make sure the syntax is correct.

correct	/ANN pdfmark
incorrect	/ann pdfmark
correct	/Rect
incorrect	/rect

3. You can separate the ingredients (arguments) with arbitrary numbers of spaces, tabs, or line-end characters.

   Following are two ways to write a pdfmark both of which are valid.

```
[ /Rect [ 75 586 456 663 ] /Contents (This is a note.) /ANN pdfmark
```

```
[
/Rect [ 75 586 456 663 ]
/Contents (This is a note.)
/ANN
pdfmark
```

4. Ingredients (arguments) can be added in any order.

   The following examples will generate the same results:

```
[
/Rect [ 75 586 456 663 ]
/Color [1 0 0 ]
/Contents (This is a note.)
/ANN pdfmark
```

```
[
/Contents (This is a note.)
/Color [1 0 0 ]
/Rect [ 75 586 456 663 ]
/ANN pdfmark
```

5. You can include comments in a pdfmark by preceding the remark with a % character. Each line of the comment must begin with the % character. Comments are helpful for documenting the use of the pdfmark.

Printing pdfmark to PostScript printers

Only PostScript interpreters designed to create PDF files can interpret pdfmarks. PostScript printers do not “understand” pdfmarks and will generate errors when they are encountered.

Put the following line at the beginning of the pdfmark recipe before the mark object ([) and the printer will ignore the pdfmark.

```
/pdfmark where {pop} {userdict /pdfmark /cleartomark load put}
ifelse
```

About the Recipes

Each recipe shows required and optional keys. Optional keys that are presented in this book cover the entries in the provided recipes. Note that additional optional keys may be available for use with some of these pdfmark types. See “References” on page 45 for additional information.

The following pages contain pdfmark recipes for these types of pdfmarks:

Annotations	/ANN
Notes	/Subtype /Text
Free Text	/Subtype /Free Text
Links	/Subtype /Link
Form Fields	/Subtype /Widget
Bookmarks	/OUT
Articles	/ARTICLES
Page Cropping	/PAGES, /PAGE
Document Information	/DOCINFO
Document View Options	/DOCVIEW
Page Labels	/PAGELABEL
Named Destinations	/DEST
Transitions	/PUT
Document Open Options	/PUT

/ANN /Text

pdfmark	Comments

`/pdfmark where {pop}`
`{userdict /pdfmark`
`/cleartomark load put}`
`ifelse`
	% Adding this line to the beginning of your pdfmark will cause a PostScript
	% printer to ignore the pdfmark when it is encountered.

`[`	% The mark object
`/Rect [72 144 288 288 ]`	% Defines the location and size of the note
`/Contents (Hello World)`	% Defines the contents of the note's comment window as Hello World
`/SrcPg 1`	% Defines the page the note is to be place on
`/Open true`	% Defines the note's comment window as open
`/Color [1 0 0 ]`	% Defines the note's color as Red
`/Title (James Kirk)`	% Defines the text in the note's title bar as James Kirk
`/ModDate (20030520)`	% Defines the date that displays in the note's title bar
`/Name Note`	% Defines the icon that is placed on the page
`/Subtype /Text`	% Defines the type of annotation as Note
`/ANN pdfmark`	% Defines the type of pdfmark as Annotation

Note Ingredients:

Required Key-Value pairs:

Key	Value	Explanation
/Rect	[X_{LL} Y_{LL} X_{UR} Y_{UR}]	Coordinates that define the open note window.
/Contents	(The popup note contents)	Contains the contents displayed in the popup window. The maximum length is 65,535 characters.

Optional Key-Value pairs:

Key	Value Example	Explanation	What happens if you do not include this key/value pair in your recipe:
/SrcPg	1	Number of the page on which to place the note.	The note will be placed on the current page.
/Open	true or false	If "true" the note will display open, if "false" the note will display closed.	The note will display closed.
/Color	[0 1 0]	An array of 3 numbers defining the percentages of Red, Green, and Blue (RGB) that define the color of the note's icon or popup window.	The note will display as grey.
/Title	(Usually the Author's Name)	Text that displays in the title bar of the popup window.	The title bar displays blank.

Key	Value Example	Explanation	What happens if you do not include this key/value pair in your recipe:

`/ModDate`	`(D:YYYYMMDD HHmmSSOHH'mm')`	Date and Time the note was last modified.
		YYYY is the year
		All other fields are optional
		MM = month (01-12)
		DD = day (01-31)
		HH = hour (00-23)
		mm = minutes (00-59)
		SS = seconds (00-59)
		Remainder defines relation to GMT. (see pdfmark reference)
			The title bar displays blank.

`/Name`	`Note`
	`Comment`
	`Help`
	`Insert`
	`Key`
	`NewParagraph`
	`Paragraph`
		The name of an icon to be displayed.	The Note icon will display by default.

`/Subtype`	`/Text`	Identifies the type of annotation as Note.	Displays a Note. Note is the default Subtype.

/ANN - /Free Text

pdfmark	comment
`[`	% mark object
`/Rect [ 36 36 288 288 ]`	% Defines the size and position of the area containing the Free Text
`/Contents (Preliminary)`	% Defines the text placed on the page
`/DA ([0 0 0] rg /Helv 18 Tf)`	% Defines the default appearance of the text % Color is Black, Font is Helvetica 18 pt.
`/BS << /W 0 >>`	% Defines the border style width as 0 (no border)
`/Q 1`	% Defines the justification as centered
`/Subtype /FreeText`	% Defines the annotation subtype as Free Text
`/ANN pdfmark`	% Defines the type of pdfmark as annotation

Free Text Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
/Rect	[X_{LL} Y_{LL} X_{UR} Y_{UR}]	Coordinates that define the open note window.
/Contents	(This is Free Text)	The text placed on the page
/DA	([0 0 0] rg /Helv 18 Tf)	default appearance, for example the color and font used. See table below for fonts that can be safely use in a pdfmark.
/Subtype	/FreeText	Identifies the type of annotation as Free Text

The following fonts can be used in the default appearance (/DA) description for a Free Text annotation:

Helvetica	= /Helv	Courier	= /Cour
Helvetica Oblique	= /HeOb	Courier Oblique	= /CoOb
Helvetica Bold	= /HeBo	Courier Bold	= /CoBo
Times	= /TiRo		
Times Italic	= /TiIt		
Times Bold	= /TiBo		

Optional Key-Value pairs:

Key	Value Example	Explanation
`/SrcPg`	`1`	Number of the page on which to place the Free Text annotation. If not included the free text will be placed on the current page.
`/Q`	`1`	defines how the text is justified. 0 = left justified 1 = centered 2 = right justified

`/BS`	`<< /S D`
	`/W 0`
	`/D [3 2] >>`
		Contains dictionary between << >> defining the border style
		/W defines border width
		0 = none
		1 = 1 pt
		2 = 2 pt
		/S defines the style
		S = Solid
		D = Dashed (/D defines dash pattern)
		B = Beveled
		I = Inset
		U = Underline
		/D defines dash pattern
		defined as an array. Default value [3]
		[3 2] defines a pattern of 3 pt dash alternating with 2 pt space

/ANN - /Links

Links can jump to other locations in the same or different pdf files. Links can also perform numerous other actions. See "Assigning Actions" on page 12 for more information on actions.

Link that jumps to the next page

pdfmark	comment
`[`	% mark object
`/Rect [72 108 144 208 ]`	% Defines the size and position of the link
`/SrcPg 2`	% Defines the page the link is placed on
`/Border [0 0 1 ]`	% Defines the border style - solid 1 pt.
`/Color (0 .5 1)`	% Defines the border color
`/Page /Next`	% Defines the next page as where the link jumps
`/View [XYZ null null 0)`	% Defines the view in this case maintain the same view
`/Subtype /Link`	% Defines the subtype as /Link
`/ANN pdfmark`	% Defines the pdfmark type as annotation

Link that jumps to page 1 and fits the view to the page width - has a green 2pt. dashed border

pdfmark	comment
`[`	% mark object
`/Rect [ 144 550 288 575 ]`	% Defines size and position of link
`/Border [ 0 0 2 [3 2] ]`	% Defines a 2 pt dashed border
`/Color [1 0 0]`	% Defines the border color as Red
`/H /I`	% Defines the highlighting style when link is clicked
`/Page 1`	% Defines the jump to page 1
`/View [ /FitH 5]`	% Defines the view to fit the width of the page in the screen
`/Subtype /Link`	% Defines the subtype as /Link
`/ANN pdfmark`	% Defines the pdfmark type as annotation

Link that launches another program

pdfmark	comment
`[`	% mark object
`/Rect [72 108 144 208 ]`	% Defines the size and position of link
`/Color (0 1 1)`	% Defines the border color as cyan
`/Action /Launch`	% Defines an action - in this case it launches another file
`/File (word.doc)`	% Defines the file opened by the launch action
`/Subtype /Link`	% Defines the subtype as /Link
`/ANN pdfmark`	% Defines the pdfmark type as annotation

Link that jumps to a web page

pdfmark	comment
`[`	% mark object
`/Rect [ 144 550 288 575 ]`	% Defines size and location of link
`/Action << /Subtype /URI /URI (http://www.jawspdf.com) >>`	% Defines an action that jumps to a web page
`/Border [ 0 0 2 ]`	% Defines the border width as 2 pt
`/Color [ 0 0 1 ]`	% Defines the color of the border as blue
`/Subtype /Link`	% Defines the subtype as /Link
`/ANN pdfmark`	% Defines the pdfmark type as annotation

Link that performs a menu item action - in this case opens Print dialog

pdfmark	comment
`[`	% mark object
`/Rect [ 50 425 295 445 ]`	% Defines the size and location of link
`/Action << /Subtype /Named /N / Print >>`	% Defined an action to activate a menu item - % in this case open the Print dialog
`/Border [ 0 0 2 ]`	% Defines the border width as 2 pt.
`/Color [ .7 0 0 ]`	% Defines the color
`/Subtype /Link`	% Defines the subtype as /Link
`/ANN pdfmark`	% Defines the pdfmark type as annotation

Link Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
/Rect	[X_{LL} Y_{LL} X_{UR} Y_{UR}]	Coordinates that define the open note window.
/Subtype	/Link	Identifies the type of annotation as Link

Optional Key-Value pairs:

Key	Value Example	Explanation
/SrcPg	1	Number of the page on which to place the link.
/Border	/Border [0 0 2 [3 2]]	The first two numbers define corner radius (0 0 is no radius). The third number defines the border width (2 = 2 pts). The second array of numbers [3 2] defines the dash pattern. The dash is 3 pts and the space is 2 pts. You do not need the second array if the line is solid.
/Color	[0 1 0]	An array of 3 numbers defining the percentages of Red, Green, and Blue (RGB) that define the color of the note's icon or popup window.

Key	Value Example	Explanation

`/H`	`/I = Invert (default)`
	`/N = none`
	`/O = outline`
	`/P = push or inset`
		Defines how the link is highlighted when clicked.

`/Page`	`2`	The page the link jumps to
`/View`	`[/Fit]`	The view of the page. See “Setting Views” on page 11 for more information.
`/Action`	`/Launch`	The action assigned to the link. See “Assigning Actions” on page 12 for more information.
`/File`	`(myfile.pdf)`	The name of the file referred to in an action

/ANN - /Widgets (forms)

A text field

pdfmark	comment
[	% mark object
/T (text input field)	% field name
/FT /Tx	% field type is text box
/DV (Summer is fun)	% Default Value is "Summer is fun"
/Rect [0 0 216 18]	% Defines field's size and location
/F 4	% Defines the field as printable
/BS << /S /S /W 1 >>	% Defines border style as a solid 1 pt line
/MK <<	% Dictionary between << >> defines form appearance
/BC [1 0 0]	% Defines the border color as red
/BG [1 1 1] >>	% Defines the background color as white
/Subtype /Widget	% Defines subtype as /Widget
/ANN pdfmark	% Defines the pdfmark as an annotation

A list box

pdfmark	comment
[	% mark object
/T (list box)	% Defines field name
/FT /Ch	% Defines field type as choice
/Rect [0 0 216 18]	% Defines field's size and location
/F 4	% Defines field as printable
/DV (Jaws PDFCreator)	% Defines the field's default value
/DA (/Helv 12 Tf 0 g)	% Defines the field's default appearance - in this case Helvetica 12 pt.
/Opt [[(Creator)(Jaws PDFCreator)] [(Editor)(Jaws PDFEditor)]]	% Defines the options available in the list [(export value) (list item)]
/Subtype /Widget	% Defines the subtype as /Widget
/ANN pdfmark	% Defines the pdfmark type as an annotation

A combo box

pdfmark	comment
`[`	% mark object
`/T (combo box)`	% Defines field name
`/FT /Ch`	% Defines field type as Choice
`/Rect [ 0 0 216 180]`	% Defines the field’s size and position
`/F 4`	% Defines the field as Printable
`/Ff 393216`	% Special flag that defines the field as a combo box
`/DV (Jaws PDFCreator)`	% Defines the default value
`/DA (/Helv 12 Tf 0 g)`	% Defines the default appearance as Helvetica 12 pt.

`/Opt [`
`[ (Creator)(Jaws PDFCreator)]`
`[ (Editor)(Jaws PDFEditor)] ]`
	% Defines the options available in the list [(export value) (list item)]

`/Subtype /Widget`	% Defines the subtype as /Widget
`/ANN pdfmark`	% Defines the pdfmark type as an annotation

A check box

pdfmark	comment
`[`	% mark object
`/T (check box)`	% Defines the field name
`/FT /Btn`	% Defines the field type as button
`/Rect [ 0 0 216 18]`	% Defines the field’s size and position
`/F 4`	% Defines the field as printable
`/BS << /S /S /W 1 >>`	% Defines the border style as 1 pt. solid line
`/MK <<`	% Dictionary between << >> defines form appearance
`/BC [ 1 0 0 ]`	% Defines border color as Red
`/BG [ 1 1 1 ] >>`	% Defines background color as White
`/Subtype /Widget`	% Defines subtype as /Widget
`/ANN pdfmark`	% Defines the pdfmark type as annotation

A Radio Button

pdfmark	comment
`[`	% mark object
`/Rect [ 261 360 297 396 ]`	% Defines the size and position of the field
`/F 4`	% Defines the field as printable
`/T (Radio)`	% Defines the name of the field
`/FT /Btn`	% Defines the field type as Button
`/DA (/ZaDb 0 Tf 0 g)`	% Defines the default appearance
`/Ff 49152`	% Special flag that defines the field as a radio button
`/Subtype /Widget`	% Defines subtype as /Widget
`/ANN pdfmark`	% Defines pdfmark type as an annotation

A Push Button

pdfmark	comment
[	% mark object
/T (button)	% Defines name of field
/FT /Btn	% Defines field type as button
/Rect [0 0 216 18]	% Defines the field’s size and position
/F 4	% Defines the field as printable
/Ff 6554	% Special flag that defines the field as a push button
/BS << /S /S /W 1 >>	% Dictionary entry that defines the border style as solid 1 pt.
/MK <<	% Dictionary entry between << >> that defines border and background color
/BC [1 0 0]	% Dictionary entry defines border color as Red
/BG [1 1 1] >>	% Dictionary entry define background color as White
/DA (/Helv 12 Tf 0 g)	% Defines default appearance - Helvetica 12 pt.
/Subtype /Widget	% Defines the subtype as /Widget
/ANN pdfmark	% Defines the annotation type as an annotation

The following pdfmark defines a dictionary that includes entries required by forms. You must include this pdfmark in any document that contains widget pdfmarks.

```
[ /_objdef {pdfDocEncoding} /type /dict /OBJ pdfmark
[ {pdfDocEncoding}
<<
/Type /Encoding
/Differences
[
24 /breve /caron /circumflex /dotaccent /hungarumlaut /ogonek /
ring
/tilde
39 /quotesingle
96 /grave
128 /bullet /dagger /daggerdbl /ellipsis /emdash /endash /florin
/fraction /guilsinglleft /guilsinglright /minus /perthousand
/quotedblbase /quotedblleft /quotedblright /quoteleft /quoteright
/quotesinglbase /trademark /fi /fl /Lslash /OE /Scaron /Ydieresis
/Zcaron /dotlessi /lslash /oe /scaron /zcaron
164 /currency
166 /brokenbar
168 /dieresis /copyright /ordfeminine
172 /logicalnot /.notdef /registered /macron /degree /plusminus
/twosuperior /threesuperior /acute /mu
183 /periodcentered /cedilla /onesuperior /ordmasculine
188 /onequarter /onehalf /threequarters
192 /Agrave /Aacute /Acircumflex /Atilde /Adieresis /Aring /AE
/Ccedilla /Egrave /Eacute /Ecircumflex /Edieresis /Igrave /Iacute
/Icircumflex /Idieresis /Eth /Ntilde /Ograve /Oacute /Ocircumflex
/Otilde /Odieresis /multiply /Oslash /Ugrave /Uacute /Ucircumflex
```

```
/Udieresis /Yacute /Thorn /germandbls /agrave /aacute /acircumflex
/atilde /adieresis /aring /ae /ccedilla /egrave /eacute
/ecircumflex /edieresis /igrave /iacute /icircumflex /idieresis
/eth /ntilde /ograve /oacute /ocircumflex /otilde /odieresis
/divide /oslash /ugrave /uacute /ucircumflex /udieresis /yacute
/thorn /ydieresis
]
>>
/PUT pdfmark
[ /_objdef {ZaDb} /type /dict /OBJ pdfmark
[ {ZaDb}
<<
/Type /Font
/Subtype /Type1
/Name /ZaDb
/BaseFont /ZapfDingbats
>>
/PUT pdfmark
[ /_objdef {Helv} /type /dict /OBJ pdfmark
[ {Helv}
<<
/Type /Font
/Subtype /Type1
/Name /Helv
/BaseFont /Helvetica
/Encoding {pdfDocEncoding}
>>
/PUT pdfmark
[ /_objdef {aform} /type /dict /OBJ pdfmark
% Define Fields array of Acroform dictionary. It will contain
entries for
% each of the widgets defined below.
% NOTE: it is not necessary to explicitly assign the widget annota-
tions
% to the Fields array; Acrobat does it automatically when the file
is opened.
[ /_objdef {afields} /type /array /OBJ pdfmark
[ {aform}
<<
/Fields {afields}
/DR << /Font << /ZaDb {ZaDb} /Helv {Helv} >> >>
/DA (/Helv 0 Tf 0 g)
/NeedAppearances true
>>
/PUT pdfmark
% Put Acroform entry in catalog dictionary
[ {Catalog} << /AcroForm {aform} >> /PUT pdfmark
```

Form Field (widget) Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
/Rect	[X_{LL} Y_{LL} X_{UR} Y_{UR}]	Coordinates that define the size and position of the field
/T	(name)	Defines the field’s name
/FT	/Tx	Defines the field’s type /Tx = Text field /Ch = Choice field /Btn = Button field
/Ff	/393216	Flag that defines the field’s subtype Choice fields: no flag = list box (leave this key value pair out) 393216 = combo box Button fields: no flag = check box (leave this key value pair out) 49152 = radio button 6554 = push button
/Subtype	/Widget	Identifies the type of annotation as Link

Optional Key-Value pairs:

Key	Value Example	Explanation
/SrcPg	1	Number of the page on which to place the field.
/BS	<< /S D /W 0 /D [3 2] >>	Contains dictionary between << >> defining the border style /W defines border width 0 = none 1 = 1 pt 2 = 2 pt /S defines the style S = Solid D = Dashed (/D defines dash pattern) B = Beveled I = Inset U = Underline /D defines dash pattern defined as an array. Default value [3] [3 2] defines a pattern of 3 pt dash alternating with 2 pt space
/MK	<< /BC [1 0 0] /BG [1 1 1] >>	Appearance characteristics dictionary that defines border and background color. Dictionary values are written between << >>.
/Opt	[[(1)(First)] [(2)(Second)] [(3)(Third)]] or [(1)(2)(3)]	Defines the options for list and combo boxes. Each array contains an export value (1) and the name that appears in the list (First). or The export value is the same as the list item name.

Key	Value Example	Explanation
`/DV`	`(Red), (2)`	The default value of the field.
`/DA`	`([0 0 0 ] rg /Helv 18 Tf)`	default appearance, for example the color and font used. See table below for fonts that can be safely use in a pdfmark.

Helvetica	= /Helv	Corurier	= /Cour
Helvetica Oblique	= /HeOb	Courier Oblique	= /CoOb
Helvetica Bold	= /HeBo	Courier Bold	= /CoBo
Times	= /TiRo		
Times Italic	= /TiIt		
Times Bold	= /TiBo		

/OUT - Bookmarks

Bookmark with formatting that jumps to a page in the same document

pdfmark	comment
`[`	% mark object
`/Page 2`	% Bookmark jumps to page 2
`/View [Fit]`	% Defines the page view to fit the page to the window
`/Title (page 2)`	% Defines the content displayed in the bookmark list
`/C [1 0 0]`	% Defines the bookmarks color
`/F 2`	% Defines the bookmarks format
`/OUT pdfmark`	% Defines the type of pdfmark as /OUT

A bookmark that jumps to a location in another file

pdfmark	comment
`[/Action /GoToR`	% Defines an action assigned to the bookmark. Jumps to another pdf.
`/File (test.pdf)`	% Defines the file the bookmark jumps to
`/Page 2`	% Defines the page in the file the bookmark jumps to
`/View [/FitR 30 648 209 761]`	% Defines the view magnification the page will display at
`/Title (Open test.pdf on page 2)`	% Defines the text that displays in the bookmark list
`/OUT pdfmark`	% Defines the pdfmark type as /OUT

A series of bookmarks that define hierarchy

pdfmarks	comments
`[/Count -3 /Page 1 /View [/XYZ 44 730 1.0] /Title (Table of Contents #2) /OUT pdfmark`	% parent pdfmark, the count key indicates the number of subordinate bookmarks. A positive count number means the bookmark is expanded to display the subordinates, a negative number means the bookmark is colapsed.
`[/Page 1 /View [/XYZ null null 0] /Title (Page 1 - Inherit) /OUT pdfmark`	% child bookmark
`[/Page 2 /View [/XYZ null null 0] /Title (Page 2 - Inherit) /OUT pdfmark`	% child bookmark
`[/Page 3 /View [/XYZ null null 0] /Title (Page 3 - Inherit) /OUT pdfmark`	% child bookmark

A bookmark that launches a file

pdfmark	comment
`[`	% mark object
`/Action /Launch`	% Defines an action assigned to the bookmark. Launches a file.
`/File (test.doc)`	% Defines the file launched by the action
`/Title (Open test.doc)`	% Defines the text that displays in the bookmark list
`/OUT pdfmark`	% Defines the pdfmark type as /OUT

A bookmark that launches to a web page

pdfmark	comment
[	% mark object
/Count 0	% Defines if the bookmark has children
/Title (Visit JAWS PDF)	% Defines the text of the bookmark
/Action /Launch	% Defines the action when the bookmark is clicked
/URI (http://www.jawspdf.com)	% Defines the web page the bookmark opens
/OUT pdfmark	% Defines the pdfmark type as an outline (bookmark)

Bookmark Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
/Title	(Table of Contents)	The text displayed in the bookmark list

Optional Key-Value pairs:

Key	Value Example	Explanation
/Count	3, -3	Indicates the number of subordinate bookmarks. A positive number means the bookmark is expanded, a negative number means the bookmark is collapsed.
/Color	[0 1 0]	An array of 3 numbers defining the percentages of Red, Green, and Blue (RGB) that define the color of the bookmark.
/F	2	Defines the format of the bookmark text 0 = normal 1 = italic 2 = bold 3 = bold italic
/Page	2	The page the bookmark jumps to
/View	[/Fit]	The view of the page. See “Setting Views” on page 11 for more information.
/Action	/Launch	The action assigned to the bookmark. See “Assigning Actions” on page 12 for more information.
/File	(myfile.pdf)	The name of the file referred to in an action
/URI	(www.jawspdf.com)	The Universal Resource Locator of a web page

/ARTICLES

Defining an article containing two beads

pdfmark	Comment
`[`	% mark object
`/Title (Now is the Time)`	% Defines the name of the article
`/Author (John Doe)`	% Defines the author of the article
`/Subject (Coming to the aid of your country)`	% Defines the subject of the article
`/Keywords (Time, Country, Aid)`	% Defines key words pertaining to the article
`/Rect [ 225 500 535 705 ]`	% Defines the size and location of the first article bead
`/Page 2`	% Defines the page the bead is on
`/ARTICLE pdfmark`	% Defines the pdfmark as an article
`[`	% mark object
`/Title (Now is the Time)`	% Defines the name of the article - same as the first bead
`/Rect [ 225 500 535 705 ]`	% Defines the size and location of the second bead
`/Page 3`	% Defines the page the bead is on
`/ARTICLE pdfmark`	% Defines the pdfmark as an article

Article Ingredients

Required Key-Value pairs:

Key	Value Example	Explanation
/Title	(Jaws PDFEditor)	The title of the article
/Rect	[X_{LL} Y_{LL} X_{UR} Y_{UR}]	Coordinates that define the size and location of the article bead

Optional Key-Value pairs:

Key	Value Example	Explanation
/Page	2	The page the article bead is on
/Subject	(Data Sheet)	The subject of the article. Only added to the first bead of the article.
/Author	(Lynn Mead)	The author of the article. Only added to the first bead of the article.
/Keywords	(Overview)	Key words relating to the article. Only added to the first bead of the article.

/PAGE, /PAGES - Page Cropping

Crop this page (crops the page where the pdfmark is placed)

pdfmark	comment
`[`	% mark operator
`/CropBox [0 0 288 288]`	% Defines the size of the cropped page
`/PAGE pdfmark`	% Defines the pdfmark type as /Page

Crop all pages (crops all pages in the document)

pdfmark	comment
`[`	% mark operator
`/CropBox [54 403 558 720]`	% Defines the size of the cropped page
`/PAGES pdfmark`	% Defines the pdfmark type as /Pages

Page Cropping Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
`/CropBox`	`[54 403 558 720]`	Defines the size of the cropped page

/DOCINFO - Document Information

pdfmark	Comment
`[`	% mark object
`/Title (My Test Document)`	% Defines the document information title field contents
`/Author (John Doe)`	% Defines the document information author field contents
`/Subject (pdfmark 3.0)`	% Defines the document information subject field contents
`/Keywords (pdfmark, example, test)`	% Defines the document information keyword field contents
`/Creator (Hand Programmed)`	% Defines the document information creator field contents
`/Producer (Jaws PDFCreator)`	% Defines the application that converted the document to pdf
`/CreationDate`	% Defines the date the document was created
`/ModDate (D:19940912205731)`	% Defines the document information modification date field contents
`/DOCINFO pdfmark`	% Defines the pdfmark type as /DOCINFO

Document Information Ingredients:

Key-Value pairs:

Key	Value Example	Explanation
`/Title`	`(My Document)`	Contents of the document information Title field
`/Author`	`(John Doe)`	Contents of the document information Author field
`/Subject`	`(Birds of the South)`	Contents of the document information Subject field
`/Keywords`	`(song birds, hawks)`	Contents of the document information Key Words field
`/Creator`	`(FrameMaker 6)`	The name of the application in which the document was created
`/Producer`	`(Jaws PDF Creator)`	The name of the application that converted the document to pdf
`/CreationDate`	`(D:19940912205731)`	The date the pdf was created. See ModDate below for format

`/ModDate`	`(D:19940912205731)`	The date the pdf was last modified
		Date and Time the note was last modified.
		YYYY is the year
		All other fields are optional
		MM = month (01-12)
		DD = day (01-31)
		HH = hour (00-23)
		mm = minutes (00-59)
		SS = seconds (00-59)
		Remainder defines relation to GMT. (see pdfmark reference)

/DOCVIEW - View Options

pdfmark	comments
`[`	% mark object
`/PageMode /UseOutlines`	% Defines the page mode display
`/Page 1`	% Defines the page the document will open to
`/View [/XYZ null null null]`	% Defines the view magnification the document opens to
`/DOCVIEW pdfmark`	% Defines the pdfmark type as /DOCVIEW

View Option Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
/PageMode	/UseOutlines	Defines how the document displays /UseNone = the document displays without bookmarks or thumbnails /UseOutlines = the document displays with bookmarks visible /UseThumbs = the document displays with thumbnails visible /FullScreen = the document displays in full screen mode

/DEST - Named Destinations

A named destination in the current document

pdfmark	comment
`[`	% Mark object
`/Dest /MyNamedDest`	% Defines the name of the named destination
`/Page 1`	% Defines the destination of the named destination
`/View [ /FitH 5 ]`	% Defines the magnification view of the named destination
`/DEST pdfmark`	% Defines the pdfmark type as /DEST

A link that will jump to the above named destination

pdfmark
`[`
`/Rect [ 70 650 210 675 ]`
`/Dest /MyNamedDest`
`/Subtype /Link`
`/ANN pdfmark`

View Option Ingredients:

Required Key-Value pairs:

Key	Value Example	Explanation
/Dest	/AName	The name of the destination

Optional Key-Value pairs:

Key	Value Example	Explanation
/Page	2	The page the destination is located on
/View	[Fit]	The magnification view of the destination

/PUT - Transitions

pdfmark	comment
[	% mark object
{ThisPage}	% Applies transition to the current page
<< /Trans	% Dictionary that defines the transition
<< /S /Blinds	% Defines the style of the transition as blinds
/Dm /H >>	% Dimension the transition is applied to (horizontal)
>> /PUT pdfmark	% Defines the pdfmark type as /PUT

pdfmark	comment
[	% mark object
{ThisPage}	% Applies the transition to the current page
<< /Trans	% Dictionary that defines the transition
<< /S /Box	% Defines the style of the transition as Box
/M /I >>	% Specifies that the transition is applied from the edges In
>> /PUT pdfmark	% Defines the pdfmark type as /PUT

pdfmark	comment
[	% mark object
{ThisPage}	% Applies the transition to the current page
<< /Trans	% Dictionary that defines the transition
<< /S /Dissolve >>	% Defines the style of the transition as Disolve
/PUT pdfmark	% Defines the pdfmark type as /PUT

pdfmark	comment
[	% mark object
{ThisPage}	% Applies the transition to the current page
<< /Trans	% Dictionary that defines the transition
<< /S /Glitter	% Defines the style of the transition as Glitter
/Di 270 >>	% Defines the direction of movement as 270°
>> /PUT pdfmark	% Defines the pdfmark type as /PUT

pdfmark	comment
[	% mark object
{ThisPage}	% Applies the transition to the current page
<< /Trans	% Dictionary that defines the transition
<< /S /R >>	% Defines the style of the transition as Replace
>> /PUT pdfmark	% Defines the pdfmark type as /PUT

pdfmark	comment
`[`	% mark object
`{ThisPage}`	% Applies the transition to the current page
`<< /Trans`	% Dictionary that defines the transition
`<< /S /Split`	% Defines the style of the transition as Split
`/D 0.5`	% Defines the duration as .5 seconds
`/M /I`	% Transition applied from the edges In
`/Dm /H >>`	% Transition applied in the horizontal dimension
`>> /PUT pdfmark`	% Defines the pdfmark type as /PUT

pdfmark	comment
`[`	% mark object
`{ThisPage}`	% Applies the transition to the current page
`<< /Trans`	% Dictionary that defines the transition
`<< /S /Wipe`	% Defines the style of the transition as Wipe
`/D 0.5`	% Defines the duration as .5 seconds
`/Di 90 >>`	% Defines the direction of movement as 90º
`>> /PUT pdfmark`	% Defines the pdfmark type as /PUT

Transition Catalogue Ingredients:

Key	Value Example	Explanation
`{ThisPage}`		Defines the place where the transition takes place. {ThisPage} = the current page {PageN} = replace N with a page number {PrevPage} = the previous page {NextPage} = the next page
`/S`	`/Split`	Defines the transition type. /Split - two lines sweep across the page either from the edges in or the center out /Blinds - several lines move across the page similar to “venetian blind” /Box - a box enlarges from the center of the page out or from the edges in /Wipe - a single line moves across the page from edge to edge /Dissolve - the old page dissolves revealing the next page /Glitter - similar to disolve but the effect sweeps from edge to edge /R - page is replaced with no effect. This is the default.
`/D`	`1.5`	Duration of the transition in seconds. The default is 1.
`/DM`	`/H`	The dimension in which the Split or Blinds transition moves. /H = horizontal /V = vertical
`/M`	`/I`	The direction of motion for Split or Box transitions /I = From the edges In /O = From the center Out
`/Di`	`270`	The direction in which the Wipe or Glitter transitions move. 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)

/PUT - Document Open Options

pdfmark	comment
`[`	% mark object
`{Catalog}`	% Defines a catalogue
`<< /ViewerPreferences`	% Catalogue entry that specifies that this is a Viewer Preference setting
`<< /HideToolbar true >>`	% Catalogue entry that defines the open option and setting
`>>`	% End of catalogue entry
`/PUT pdfmark`	% Defines the pdfmark type as /PUT

View Option Ingredients:

Key-Value pairs:

Key	Value Example	Explanation
/HideToolbar	true or false	This will hide the toolbars. Default is false
/HideMenuBar	true or false	This will hide the menu bar. Default is false
/HideWindowUI	true or false	This will hide User Interface elements such as scrollbars Default is false
/FitWindow	true or false	Fits the window to the file. Default is false
/CenterWindow	true or false	Centers the window in the screen Default is false
/Page Layout	/SinglePage /OneColumn /TwoColumnLeft /TwoColumnRight	Displays one page at a time Displays in continuous page mode Displays continuous facing pages binding on left Displays continuous facing pages binding on right
/NonFullScreenPageMode	/UseNone /UseOutlines /UseThumb	Displays the page only Displays the page and bookmarks Displays the page and thumbnails

How to implement pdfmarks

Placing pdfmarks in the PostScript Code

Placed by the application automatically

Many applications, such as FrameMaker or MSWord, are now supporting the automatic generation of certain pdf features. These applications use pdfmarks to generate these features but the process is transparent to the user. If your application is already set up to generate the features you want, it will be a much easier to let the application do it rather than writing pdfmarks to accomplish the task yourself.

Application generated pdf features can include:

- Bookmarks that jump to a location in the document
- Note annotations
- Document Open settings
- Hypertext Links
- Articles
- Document Information

Placed manually into the application

There are still things that can be accomplished with manually created pdfmarks that are not yet supported directly by applications. Some applications provide ways for the user to manually insert pdfmarks or you can create eps files that contain pdfmarks that can be placed in any application that supports the eps format. Following are brief description of how to enter pdfmarks into FrameMaker Postscript text frames, into MSWord using field codes and instructions for creating pdfmarks as eps files.

Using PostScript text frames in FrameMaker

1. Draw a PostScript text frame in FrameMaker.
2. Enter your pdfmark in the text frame.
3. Select the text frame and select Graphics > Object Properties from the menu
4. In the Object Properties dialog select the PostScript Code checkbox and click OK.

The text field is now defined as postscript code and is not editable. The contents will be included in the PostScript stream when the file is printed. To edit the field you must deselect the PostScript code check box in the Object Properties dialog.

Additional Information:

- Remember, if you want to print your file to a PostScript printer add the following line to your pdfmark:
```
/pdfmark where {pop} {userdict /pdfmark /cleartomark load put} ifelse
```
- In FrameMaker the PostScript text frame provides coordinates for the frame. To use the coordinates of the Postscript text frame as the coordinates for the /Rect key insert the following as the value of the key:
```
/Rect [ 7 -4 roll 4 -2 roll pop pop 0 0 ]
```
- If the pdfmark does not contain the /Rect key, you should get rid of the PostScript text frame coordinates by entering the following at the beginning of the pdfmark.
```
pop pop pop pop
```

See “References” on page 45 for additional references that cover using FrameMaker PostScript text frames.

Using Field Codes in MS Word

1. Select Tools > Options from the menu and select the Field Codes check box. This makes the field codes visible while you are working.
2. Select Insert > Fields from the menu. Select Print from the list of field names.
3. Enter the pdfmark between quotes. Note: Do not insert line brakes, commas, or full stops as these will cause errors.
4. The pdfmark will be inserted into the PostScript stream when the file is printed.

Additional Information:

- Remember, if you want to print your file to a PostScript printer add the following line to your pdfmark:
```
/pdfmark where {pop} {userdict /pdfmark /cleartomark load put} ifelse
```
- You can use the \p para options to define a variable that uses the current paragraph’s boundaries as the coordinates for the pdfmark. Example:
```
{print \p para “[/Rect [ wp$box pathbbox ] /Page 2 /Subtype /Link /ANN pdfmark”}
```
- You can use the \p cell option to define a variable that uses the table cell’s boundaries in which the Field code is entered as the coordinates for the pdfmark. Example:
```
{print \p cell “[/Rect [ wp$box pathbbox ] /Page 2 /Subtype /Link /ANN pdfmark”}
```

See “References” on page 45 for additional references with information about using field codes.

EPS files

Placing pdfmarks inside an EPS file

1. Open a text editor (Notepad for instance).

2. Enter the following text:

   ```
   %!PS-Adobe-3.0 EPSF-3.0
   %%BoundingBox: 0 0 216 18
   systemdict /pdfmark known not {userdict /pdfmark systemdict /
   cleartomark get put } if
   ```

   The four numbers after the BoundingBox: statement are coordinates that define the size of the eps. These coordinates should match the coordinates of any /Rect key value in the pdfmark.

3. Enter your pdfmark after this text.

4. End with the following text after your pdfmark:

   ```
   %EOF
   ```

5. Save the file as a **text file** with an .eps extension.

Example of a EPS with a link:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 216 18
systemdict /pdfmark known not {userdict /pdfmark systemdict /
cleartomark get put } if
[
/Rect [0 0 216 18 ]
/SrcPg 2
/Border [0 0 1 ]
/Color (0 .5 1)
/Page /Next
/View [XYZ null null 0)
/Subtype /Link
/ANN pdfmark
%%EOF
```

Additional Information:

- You can place this EPS in any application that supports the EPS format.
- The EPS will display as a gray box because it does not contain a preview.
- The pdfmark will use the size and position of the EPS as it’s coordinates.

pdfmarks in EPS graphics

1. Create your graphics in an illustration application that can save an EPS.

2. Define the bounding box of your graphic by drawing a rectangle with the exact dimensions as the desired rectangle is to appear in the pdf. The stroke and fill properties should be none.

3. Position your graphic within the rectangle.

4. Change the ruler origin to be the lower-left corner of the rectangle.

5. Save the graphic as an EPS.

6. Open the EPS in a text editor.

7. Locate the following section:

```
%%PageTrailer
gsave annotatepage grestore showpage
%%Trailer
```

8. Insert your pdfmark code below the %%PageTrailer line.

9. Edit the values of the /Rect key to match the size of the rectangle you drew to define the bounding box size of the EPS.

10. Save your edited EPS file and it's ready to place in your documents.

Additional Information:

- If you open an EPS that contains pdfmarks in the illustration program and save it you will lose the pdfmarks.
- If your application will not place the EPS try saving it without a preview.

Prologue and Epilogue

Placing pdfmarks in the Prologue and/or Epilogue files

PDFCreator

1. Locate the prologue file in the following default location: \Jaws Systems\JawsPDFCreator\Default\prologue.ps
2. Make a copy of this file and name it prologue.bak
3. Open the prologue file with a text editor such as Notepad.
4. Add your pdfmarks at the end of this file and save. Make sure you re-save it as a text file.
5. Create your pdf from within any application by printing to the PDFCreator printer. The pdfmarks in the prologue are added to the PostScript stream and are converted to pdf features.

Additional Information:

- The pdfmarks placed in the prologue file will be added to every pdf you create until you remove them from the prologue (restore the original from the backup copy).

Acrobat Distiller

1. Locate the prologue and epilogue files in the following default location: \Adobe\Acrobat X.0\Distillr\Data\prologue.ps (or epilogue.ps)
2. Make copies of these files as backups (prologue.bak, epilogue.bak)
3. Open either the prologue.ps or epilogue.ps files and enter your pdfmark after the line that says, "%Insert your custom PostScript here..."
4. Save the file. Make sure you save it as a text file.
5. Open Distiller and select the Job Option you intend to use. Select Settings > Job Options from the menu.
6. In the Job Options dialog select the Advanced tab and select the "Use Prologue.ps and Epilogue.ps" check box. Rename the job option if necessary and save.
7. Distilling files using this job option will process the pdfmarks you placed in the prologue.ps and epilogue.ps as the job is distilled.

Additional Information:

- Both the prologue and the epilogue must be in the default directory, even if you have only used one of the files to enter your pdfmarks.

- If the Job Option you choose does not have the “Use Prologue.ps and Epilogue.ps” selected then your pdfmarks will not be processed.
- If the Job Option you choose has the “Use Prologue.ps and Epilogue.ps” option selected, then every file processed with this Job Option will have your pdfmarks applied.

References

Adobe PDFMark Reference Manual - Technical Note #5150
http://partners.adobe.com/asn/developer/acrosdk/docs/createpdfapi
/pdfmarkReference.pdf

Adobe PDF Specification version 1.4
http://partners.adobe.com.asn.acrobat.docs.js#filefmtspecs

PDFMark Primer, chapter 6 of “Web Publishing with Acrobat/pdf” by Thomas Merz available as pdf from:
http://www.planetpdf.com

PDFmark Boiler-Plate Template by Michael Little. This template contains 'boilerplate field codes to insert PDFmarks into MS Word Documents available from:
http://www.planetpdf.com

FrameMaker to Acrobat courses - by Shlomo Perets
http://www.microtype.com