PhUSE EU Connect 2018

Paper AD07

Post-Processing PDFs with Ghostscript

Jonathan North, S-cubed Biometrics Ltd, Abingdon, UK

ABSTRACT

Creating and manipulating PDFs is a common task in the creation of CDISC compliant submissions, and in the combining of TFLs into a single file. These documents are more usable if they contain bookmarks and annotations. Commercial tools exist that allow the creation of bookmarks and annotations; but for the most part are quite cumbersome, requiring a 'click-heavy' work-flow. Considering that a lot of the meta data for the creation of bookmarks and annotations will be already known, or easily created, a programmatic method of creating bookmarks and annotations is desirable. Ghostscript is a suite of software utilities that includes a command line tool that can add annotations and bookmarks to a pre-existing PDF based on a simple mark-up language. This paper intends to introduce some of the ways the Ghostscript tools can be used to manipulate a PDF file, including creation of bookmarks, annotations, and setting document properties.

INTRODUCTION

Past PhUSE papers have been presented giving examples of how to post-process or otherwise manipulate PDF files, including by the author (see [1] and [2]), however there are limitations in the methods presented. The FDA has strict requirements on what is allowed in a submission PDF document (see [3]); in particular, the requirement that there be no JavaScript embedded in the PDF document. Both the referenced papers make use of JavaScript to create bookmarks, and therefore are not suitable for submission, even if they are OK for internal company use. This paper will explore an alternate method of creating bookmarks, using the open source tool Ghostscipt. However, Ghostscript provides more functionality that just creating bookmarks, some of which this paper will explore.

Ghostscript is an open source application that has been under development for 20+ years (see [4]). It is a command line tool that makes it ideal for integrating into a development environment, to allow post-processsing of PDF files.

In its most basic usage Ghostscript can be used to combine or split existing PDF files. However, by supplying a secondary file containing a list of pdfmarks it is possible to do much more. This paper will present some of the functionality that Ghostscript provides.

- Combining and splitting PDFs
- Using pdfmarks to
  - Adding annotations to a PDF, e.g. for use in an annotated CRF
  - Adding bookmarks to a PDF
  - Setting PDF document properties, e.g. initial page to open to, default view, page magnification
- Adding a watermark to a PDF

Note that when discussing the use of a pdfmarks file there is no detail of how best to create this file. The pdfmarks file is a simple text file, and as such can be created manually or programmatically. Programmatic creation will provide the most benefit, however, the language used to create the file will depend on where the source metadata is maintained. As examples, the file could be created in SAS, or in VBA embedded into an Excel file. Both SAS and Excel VBA could be used to run Ghostscript with the necessary command line, providing further automation.

SIMPLE COMMAND LINE USAGE

Ghostscript is a command line tool, and provides a lot of functionality that is controlled by specifying one or more switches. The command line call will take the following format:

```
gs [options] {filename 1} ... [options] {filename N} ...
```

Note:

- `gs` here should be replaced with the correct command line based on your installation, and if necessary specify the full path to the command. For example, if Ghostscript is installed into the top-level of C:\ on a Windows PC the command would be similar to `c:\Ghostscript\bin\gswin64.exe`
- In a similar manner, depending on your current working directory it may be necessary to specify the full path of input and output files. To keep the following examples as simple as possible the full path will be ignored.

The most common options when working with PDFs are:

`-dBATCH`	Causes Ghostscript to exit once it has finished processing
`-dNOPAUSE`	Causes Ghostscript to process all pages without pausing at the end of each page
`-dQUIET`	Suppress routine information comments
`-sOutputFile=`	Specify the output file to create

Note:

- to make the following examples shorter, the `-dQUIET -dBATCH -dNOPAUSE` switches will not be shown.

COMBINING MULTIPLE FILES

Combining multiple files into a single PDF is perhaps the easiest thing to do with Ghostscript. In a reporting environment each TFL might be produced as a single PDF file. If this is the case, then it will often be desirable to combine them into a single file. In order to do this the following command would be executed.

```
gs -sDEVICE=pdfwrite -sOutputFile=combined.pdf file1.pdf file2.pdf file3.pdf
```

SPLITTING A PDF, OR EXTRACTING SPECIFIC PAGES

When extracting one or more pages from a PDF there are several variants that can be employed.

- Extracting a continuous block of pages
- Extracting a non-continuous block of pages, and in an alternate order
- Extracting a block of pages, one page per output file

Extracting a block of pages

```
gs -sDEVICE=pdfwrite -dFirstPage=2 -dLastPage=3 -sOutputFile=subset.pdf combined.pdf
```

Extracting a non-continuous block of pages

There is more flexibility in this method in that there are a number of options in how to specify the pages

`-sPageList=odd`	odd pages are selected
`-sPageList=even`	even pages are selected
`-sPageList=1,3,5-8`	a random selection of pages, note that selected pages must be in order

```
gs -sDEVICE=pdfwrite -sPageList=even -sOutputFile=pagelist_even.pdf combined.pdf
```

Extracting a block of pages, one page per file.
Any of the above methods of page selection can be used to define the pages to extract; in order to extract one page per file the output filename should include %d as part of the output file name. Note, however, that the page number does not match the source page; e.g. in the following example, even though the page range 2-3 is specified the output files will be subset1.pdf and subset2.pdf

```
gs -sDEVICE=pdfwrite -dFirstPage=2 -dLastPage=3 -sOutputFile=subset%d.pdf combined.pdf
```

USING PDFMARKS TO CUTOMISE A PDF

A PDF can be processed in more complex ways by using pdfmarks. This allows to add enhancements to the PDF such as annotations and bookmarks, or to set various document properties. All pdfmarks are simple text and are created in a text file that is then specified in the command line call. It is possible to specify multiple pdfmarks in the same file.

Once the pdfmarks file has been created, the following is an example of how to instruct Ghostscript to read and interpret the required pdfmark enhancements.

```
gs -sDEVICE=pdfwrite -sOutputFile=processed.pdf -dPDFSETTINGS=/prepress pdfmarks.txt
original.pdf
```

WHAT IS A PDFMARK

pdfmarks are extensions to the PostScript language that are used to represent PDF features, such as annotations and bookmarks. They have a strict syntax in how they are defined, and are made up of three parts.

`[`	The `[` is used to mark the start of a new pdfmark
`Arguments`	used to describe the features for the current pdfmark
`A name`	the kind of pdfmark being described

As an example, the following defines a bookmark

`[`	Mark the start of a new pdfmark
`/Page 1`	Multiple arguments
`/Title (Bookmark Title)`	
`/OUT pdfmark`	Define the pdfmark name

As can be seen in this example, multiple arguments can be specified. White space and carriage returns have no meaning in the pdfmark syntax. The following has exactly the same meaning as the above

```
[/Page 1 /Title (Bookmark Title) /OUT pdfmark
```

Documentation of the pdfmark syntax is available online [5].

CREATING ANNOTATIONS WITH PDFMARKS

PDF annotations are of most use in creating an annotated CRF. The CDISC Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG) provide guidance on what to annotate along with suggestions for the text to use, in compliance with FDA Study Data Specifications. These guidelines suggest using different size and colour fonts to distinguish between Domain and Variable annotations. In addition, if there are multiple Domains represented on the CRF page, it is suggested that each have a different background colour. (Figure 1)

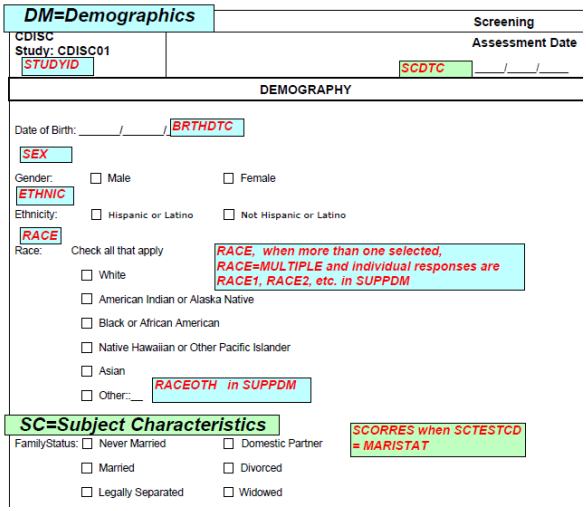

Figure 1 Suggested formatting of annotations in SDTM-MSG

The following pdfmark defines a Domain style annotation

```
[
/SrcPg 1                                Define the page to create the annotation
/Rect [100 400 300 430]                 Define the bounding box of the annotation
/Subtype /FreeText                      Define the type of annotation
/Color [1 1 0.75]                       Define the background colour, in this case cream
/DA ([0 0 0] rg /Cour 12 Tf)            Define the text colour and font, in this case black Courier 12pt
/Contents (Test Dataset Annotation)     Define the text for the annotation
/ANN pdfmark
```

The following pdfmark defines a Variable style annotation

```
[
/SrcPg 1                                Define the page to create the annotation
/Rect [100 450 300 480]                 Define the bounding box of the annotation
/Subtype /FreeText                      Define the type of annotation
/Color [1 1 0.75]                       Define the background colour, in this case cream
/DA ([1 0 0] rg /Cour 8 Tf)             Define the text colour and font, in this case red Courier 8pt
/Contents (Test Variable Annotation)    Define the text for the annotation
/ANN pdfmark
```

Multiple line annotations can be created by including a \n in the annotation text, as in the following example

```
[
/SrcPg 1                                Define the page to create the annotation
/Rect [100 500 300 530]                 Define the bounding box of the annotation
/Subtype /FreeText                      Define the type of annotation
/Color [1 1 0.75]                       Define the background colour, in this case cream
/DA ([1 0 0] rg /Cour 8 Tf)             Define the text colour and font, in this case red Courier 8pt
/Contents (Annotation\nOver 2 lines)    Define the text for the annotation
/ANN pdfmark
```

These three annotations pdfmarks will create the following in the PDF. (Figure 2)

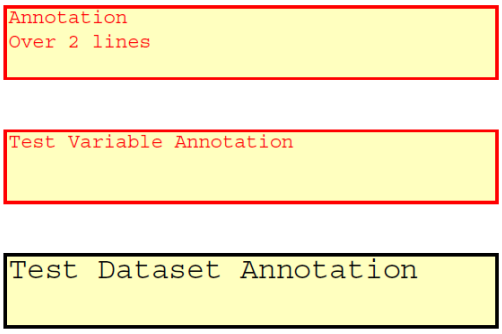

Figure 2 Annotations created with pdfmarks

In a PDF the page dimensions are defined in points, with the origin as the lower-left corner of the page. Bounding boxes for annotations are defined with the `/RECT[`X_{LL} Y_{LL} X_{UR} Y_{UR}`]` argument, and specify the lower-left and upper-right corners of the bounding box. The actual size of the bounding box should be determined by the text content. If a monospace font is used, such as Courier in the examples, it is possible to calculate the required bounding box size – but that is outside the scope of this paper.

Colours are defined as RGB values with each component specified in the range 0-1, e.g `[1 0 0]` defines red.

CREATING BOOKMARKS WITH PDFMARKS

To aid in navigation in a PDF bookmarks can be added. These would normally be created to match with section and sub-section heading in the document, but can also be created to link between PDF documents, or even to web-pages. Bookmarks can be created with a nested hierarchy, and if so, created either closed or open. (Figure 3)

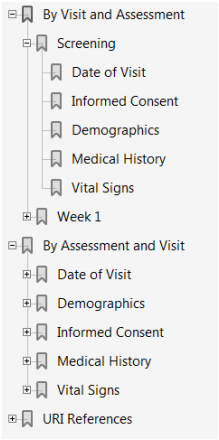

Figure 3 Example of PDF bookmarks

A simple top-level bookmark

```
[
/Title (Date of Visit)                    Define the bookmark text
/Page 1                                   Define the page to link to
/OUT pdfmark
```

A top-level bookmark, with two child bookmarks

```
[
/Title (By Visit and Assessment)          Define the bookmark text
/Count 2                                  Define the number of child bookmarks
/OUT pdfmark
[/Title (Screening) /Page 1 /OUT pdfmark
[/Title (Week 1) /Page 2 /OUT pdfmark
```

The above example will create the “Screening” bookmark as open. If instead the bookmark should be created closed, then the count should be specified as a negative number.

A bookmark to a different PDF file

```
[
/Title (New PDF File)
/Action /Launch /File (new_pdf_file.pdf)
/OUT pdfmark
```

A bookmark to a web-page

```
[
/Title (pHuse)
/Action /Launch /URI (https://www.phuse.eu/)
/OUT pdfmark
```

SETTING PDF DOCUMENT PROPERTIES WITH PDFMARKS (HEADER 2)

There are a number of settings in a PDF document that determine how the file first opens. These include such things as the initial page to show, the page magnification, whether to show one of the sidebars (e.g. bookmarks, thumbnails, etc.), and various document properties. All of these can be set using pdfmarks as the following examples show.

To set the page display option

```
[
{Catalog} << /PageLayout /SinglePage >>          Set the document to open showing a single page
/PUT pdfmark
```

To set the initial page to show, whether to show a sidebar, and page magnification

```
[
/PageMode /UseOutlines          Set the document to open showing bookmarks
/Page 1                         Set the document to open showing page 1
/View [/Fit]                    Set the document to open scaled to fit the window
/DOCVIEW pdfmark
```

To set one or more document properties

```
[
/Title (PhUSE Paper)
/Author (Author Name)
/Keywords (PhUSE, Application Development, PDF)
/DOCINFO pdfmark
```

MORE COMPLEX PDF PROCESSING

Ghostscript can process more than just PDF files, with pdfmarks for adding enhancements. In addition, Ghostscript can work with the PostScript language directly. The PostScript language is beyond the scope of this paper, but as a programming language in its own right it can be used to describe the appearance of text, graphical shapes, and sampled images on printed or displayed pages.

Ghostscript makes use of PostScript files in the same way that pdfmarks files are used. That is, the following command can be used

```
gs -sDEVICE=pdfwrite -sOutputFile=processed.pdf -dPDFSETTINGS=/prepress postscript.ps
original.pdf
```

As an example of what can be achieved in this way, the following code will add a watermark to a PDF.

```
/CompatibilityLevel 1.4
/watermarkText { ( TEXT TO USE IN WATERMARK ) } def
/watermarkFont { /Helvetica-Bold 72 selectfont } def
/watermarkColor { .75 setgray } def
/watermarkAngle { 45 } def

/pageWidth { currentpagedevice /PageSize get 0 get } def
/pageHeight { currentpagedevice /PageSize get 1 get } def

<<
/EndPage {
0 .pushpdf14devicefilter % depth .pushpdf14devicefilter -
2 eq { pop false }
{
gsave
<< >> clippath pathbbox newpath .begintransparencygroup
.3 .setopacityalpha 1 .setshapealpha
watermarkFont
watermarkColor
pageWidth .5 mul pageHeight .5 mul translate
0 0 moveto
watermarkText false charpath flattenpath pathbbox
4 2 roll pop pop
0 0 moveto
watermarkAngle rotate
-.5 mul exch -.5 mul exch
rmoveto
watermarkText show
.endtransparencygroup
grestore
true
} ifelse
.poppdf14devicefilter
} bind
>> setpagedevice
```

CONCLUSION

PDF files form an integral part of a submission. Being able to easily post-process the PDF files adding enhancements to them makes the PDF files that much more useable by the reader. By using Ghostscript and its ability to process pdfmarks files a great deal of functionality can be added to otherwise simple PDF files. While not described in this paper, by creating the pdfmarks file programmatically, much of this functionality can easily be added into a development workflow.

REFERENCES

[1] CC04(2015): Automating Production of the blankcrf.pdf in a CRO Environment,Jonathan North, S-cubed Biometrics.
[2] TS06(2015): Unleash the Power of Adobe Acrobat® using JavaScript
[3] Portable Document Format (PDF) Specifications (September 2016, Version 4.1)
https://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM163565.pdf
[4] Ghostscript FAQ https://www.ghostscript.com/faq.html
[5] pdfmark Reference https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/pdfmark_reference.pdf

RECOMMENDED READING

Cooking up Enhanced PDF with pdfmark Recipes.
http://www.meadowmead.com/wp-content/uploads/2011/04/PDFMarkRecipes.pdf
This online reference provides several useful examples, along with more detail on the more commonly used components of the pdfmarks syntax. This provides a simpler introduction to using pdfmarks, before referring to the Adobe pdfmark reference manual [5].

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jonathan North
S-cubed Biometrics Ltd
99 Park Drive
Milton Park
Abingdon
Oxon
OX14 4RY
UK
Email:jn@s-cubed.co.uk
Web: http://www.s-cubed-global.com/