Calling Java from SAS for PDF processing

Phuse SDE, 10-10-2019, Utrecht

Disclosure statement

Rob Wartenhorst is an employee of the GSK group of companies. This work was sponsored by GlaxoSmithKline Biologicals SA.

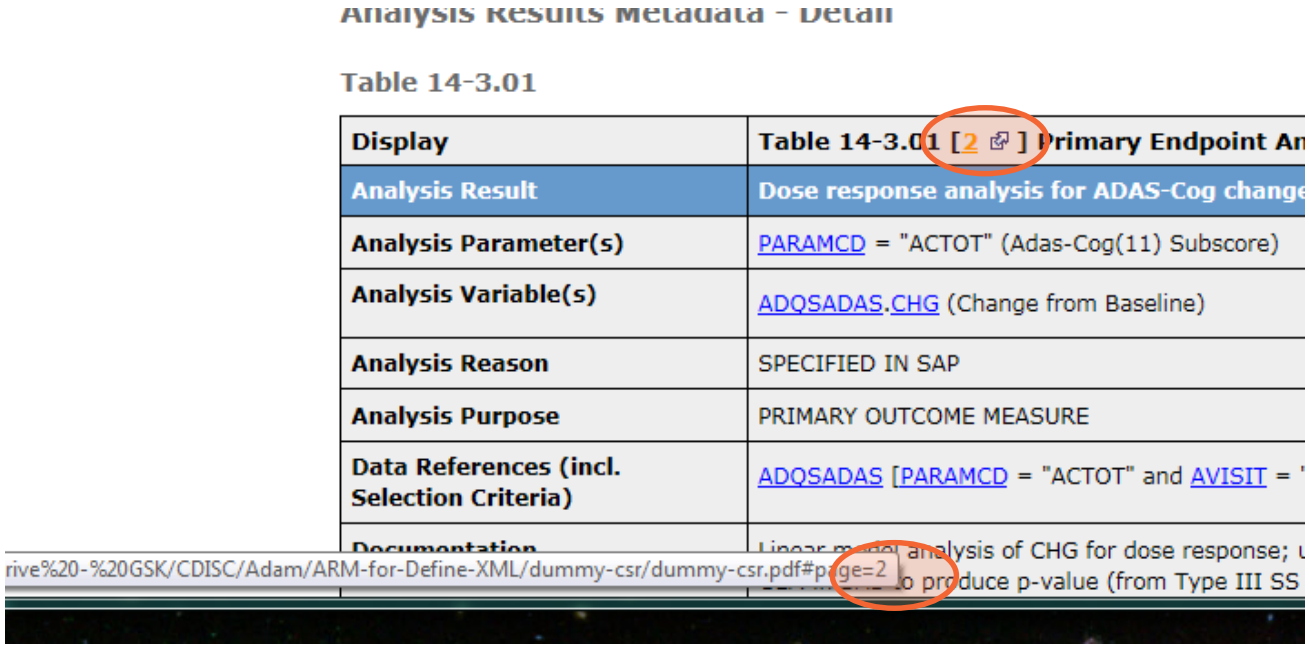
Agenda

- Objectives
- Initial approach
- New approach using Java tool

Objectives

Objectives

- Combine all CSR SAS output in a single bookmarked PDF
- Automated hyperlinking to this bookmarked PDF from the DEFINE.XML



Initial approach

Initial approach

1. Generate TFL output

– Create each output in PDF format and also create an itemstore file (extension sas7bitm) using ODS DOCUMENT

```
** Set-up ODS DOCUMENT section ;
ODS DOCUMENT NAME=sas7bitm.&outfile2(UPDATE) DIR=(PATH=\&outfile2(WRITE) LABEL="&outfile");
```

– PROC report;

……

```
3
4 ODS DOCUMENT CLOSE;
5
```

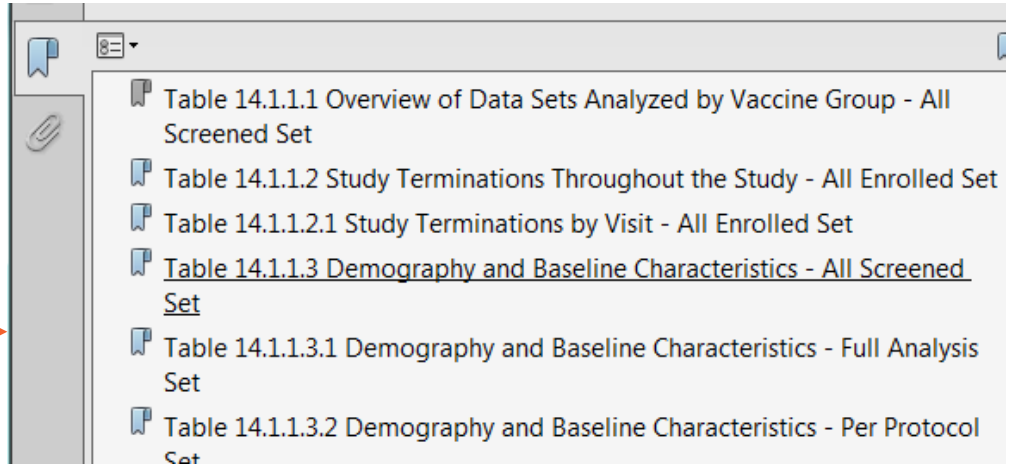
Initial approach

2. Combine all outputs in a single PDF

- Again, using ODS DOCUMENT, replay all the generated outputs
  - Rearrange bookmarks from multi-level to single-level
  - Then regenerate the output into a single PDF with bookmarks

```
455   ODS LISTING CLOSE;
456
457   ODS PDF FILE=OUTFILE STYLE=&style;
458   PROC document NAME=work.contain;
459     REPLAY ^;
460   RUN;
461   QUIT;
462
463   ODS PDF CLOSE;
464   ODS LISTING;
```

SAS generated PDF with bookmarks



Initial approach

Some issues…

- Page numbering changes from ‘page x of y’ for the individual tables to ‘page x of y’ of the complete package
- Output looks different if a different style is active (e.g. different style for graph than text)
- Output looks different if the orientation is different than in the original run
- Output looks different if the margines are different than in the original run
- In SAS 9.4 there seems to be a bug that if you change orientation in between replay steps, the margines set are not correctly respected
- When using {lastpage} notation, bug in SAS that causes page to be a black box after changing page orientation
- Replaying a lot of outputs would cause SAS to run out of memory

Can we use Java tools?

Java Tools

What can you do with Java

- In LSAF you can't call system functions through an "X" command, but you can call Java functions
- SAS has many API's in LSAF that use Java
- In LSAF we have progressively developed a custom API toolbox that is a java library. Examples of some tools:
  - Archives management
  - Checksum
  - Versioning management
  - Subscription management
  - Audit management
  - **PDF functionalities**
  - …

Java Tools

SAS example

```
%LET newclass=&newclass.&_sep.&sddshare./SDDEXTAPI/JAR/sdd4_ext_macros.jar;
%LET newclass=&newclass.&_sep.&sddshare./SDDEXTAPI/JAR/sdd-file-toolkit.jar;
OPTIONS SET = CLASSPATH "%NRBQUOTE(&newclass)";

%MACRO sdd_mergepdfs (pdfin1=, pdfin2=, pdfout=)
   * Call to Java ;
 DATA _null_;
  LENGTH result_as_text debug grmsg pdfin1 pdfin2 pdfout $ 4096 ;
  DECLARE JAVAOBJ j ( 'com/sdd/toolkit/FileToolkit' );

   pdfin1 = STRIP(SYMGET( 'pdfin1' ));
   pdfin2 = STRIP(SYMGET( 'pdfin2' ));
   pdfout = STRIP(SYMGET( 'pdfout' ));

   j.callStringMethod( 'mergePdf', STRIP(pdfin1), STRIP(pdfin2), STRIP(pdfout), grmsg );
   CALL SYMPUTX( '_sddmergepdfs_', STRIP( grmsg ) );
  RUN;

%MEND;
```

Java Tools

Why Java ?

- Through the Java API provided by LSAF, a Java program can use all features provided by the LSAF GUI. This program can be run :
  - Inside a SAS program, using the connected user's session
  - Directly from the workstation, by providing connection parameters

- Any of the many existing Java libraries can be added to that Java program, instantly providing a wide array of advanced functionalities. For example Apache PDFBOX

Java Tools

What can Apache PDFBox do?

- Open source tool Apache PDFBox (https://pdfbox.apache.org/)
  - Decrypt/Encrypt
  - Extract text/images
  - overlayPDF
  - PDFDebugger
  - **Merge PDF**/Split PDF
  - PDFSplit
  - PDFToimage
  - PrintPDF
  - TextToPDF
  - Plus many more smaller functions, including access to **bookmarks**

Java Tools

The tool appllied for our use case

- Merge PDF files
  - Simple call of the Mergepdf class (see slide 11)
- Retrieve bookmark information

Tool to extract the bookmarks

The challenge

- The Java function to extract the bookmarks structure is not very complex: a recursive function going down the bookmark tree, building a string as it goes
- The real challenge lies in the interface between the SAS program and the Java calls
  - Only basic types are supported in the Java signature (number, string…)
  - Java functions that “do” something can return a status code
  - Java functions that read something (e.g. PDF bookmarks, object audit trail…) need a more complex interface
- In the case of the PDF bookmarks, the Java return is a string of a predefined format (separators), that is then parsed and converted in a dataset
  - **Limitation**: SAS can only accept a string up to 32767 characters

Tool to extract the bookmarks

SAS example

```
%LET newclass=&newclass.&_sep.&sddshare./SDDEXTAPI/JAR/sdd4_ext_macros.jar;
%LET newclass=&newclass.&_sep.&sddshare./SDDEXTAPI/JAR/sdd-file-toolkit.jar;
OPTIONS SET = CLASSPATH "%NRBQUOTE(&newclass)";

%MACRO sdd_pdfbookmark (localfile=)
    * Call to Java ;
  DATA _null_;
    LENGTH result_as_text $ 32767 ;
    localfile = STRIP(SYMGET('localfile'));

    DECLARE JAVAOBJ j ( 'com/sdd/toolkit/FileToolkit' );
    j.callStringMethod( pdfBookmarks', STRIP(localfile), result_as_text);
    CALL SYMPUTX( 'bookmarks', STRIP( result_as_text ) );
   RUN;

%MEND;

%sdd_pdfbookmark(localfile = &_sasws_/mybookmarkedpdf.pdf);
```

Tool to extract the bookmarks

Key parts of the java code and sample result

```
public String pdfBookmarks(String localFile) {
String newLine = "{/l}";
String sep = "{/c}";

ret = "0"+sep+"#ttp#"+sep+pddoc.getNumberOfPages()+newLine+ret;
```

Sample result:
_sdd_pdfresult=0{/c}#ttp#{/c}348{/1}{/c}Table 14.1.1 Summary of disposition{/c}1{1{/1}1{c}
Table 14.1.2 Summary of demography{/c}2{/1}1{/c}14.1.3 Summary of medical history{/c}4{/1}

Reference: https://memorynotfound.com/apache-pdfbox-bookmark-pdf-example/

New approach

Using SAS & Java tools

New approach using SAS & Java tools

1) First step, creating a single table in PDF format

– Create each output in PDF format and also create an itemstore file (extension sas7bitm) using ODS DOCUMENT

```
** Set-up ODS DOCUMENT section ;
ODS DOCUMENT NAME=sas7bitm.&outfile2(UPDATE) DIR=(PATH=\&outfile2(WRITE) LABEL="&outfile");
```

– PROC report;

.....

```
3
4 ODS DOCUMENT CLOSE;
5
```

– Rearrange bookmarks from multi-level to single-level. Reference paper: https://www.pharmasug.org/proceedings/2018/QT/PharmaSUG-2018-QT12.pdf

– Then replay the output into a PDF with proper bookmark

```
455   ODS LISTING CLOSE;
456
457   ODS PDF FILE=OUTFILE STYLE=&style;
458   PROC document NAME=work.contain;
459     REPLAY ^;
460   RUN;
461   QUIT;
462
463   ODS PDF CLOSE;
464   ODS LISTING;
```

New approach using SAS & Java tools

2) Bundle all PDF's and extract bookmarks

- Using logic from slide 11. Then:

  %sdd_mergepdfs(pdfin1=pdffile1.pdf, pdfin2=pdffile2.pdf, pdfout = bundle.pdf);

  %sdd_mergepdfs(pdfin1=bundle.pdf, pdfin2=pdffile3.pdf, pdfout = bundle.pdf);

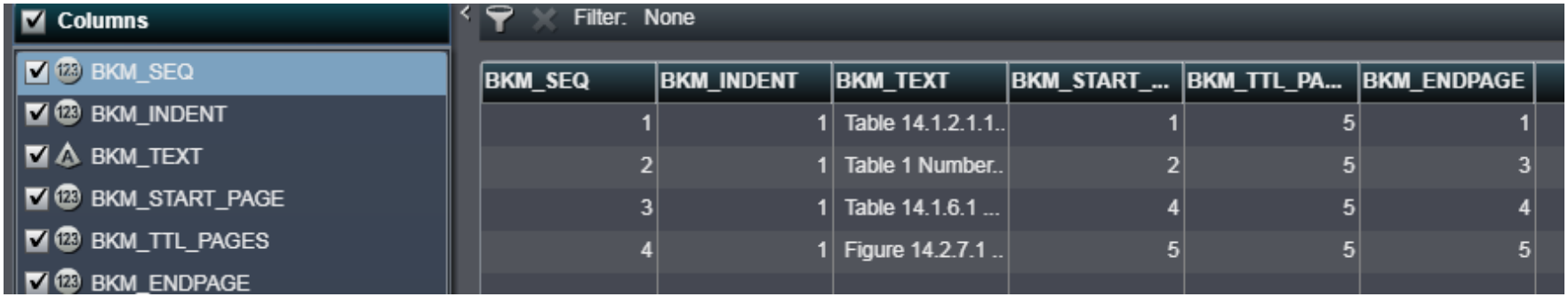
  …

  %sdd_mergepdfs(pdfin1=bundle.pdf, pdfin2=pdffile#.pdf, pdfout = **bundle.pdf**);

- Using logic from slide 16. Then:

  %sdd_pdfbookmark(localfile = &_sasws_/**bundle.pdf**);

- Resulting dataset:

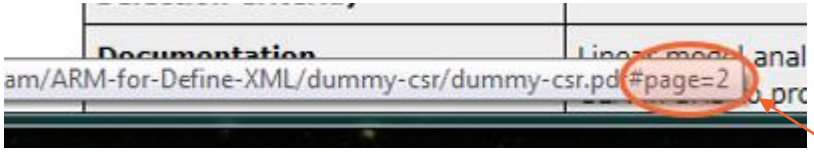


BKM_SEQ	BKM_INDENT	BKM_TEXT	BKM_START_...	BKM_TTL_PA...	BKM_ENDPAGE
1	1	Table 14.1.2.1.1..	1	5	1
2	1	Table 1 Number..	2	5	3
3	1	Table 14.1.6.1 ...	4	5	4
4	1	Figure 14.2.7.1 ..	5	5	5

Steps in generating output

3) Generate DEFINE.XML

– Program generating DEFINE.XML reads all the metadata, now also including the page number for each output table



Columns

- BKM_SEQ
- BKM_INDENT
- BKM_TEXT
- BKM_START_PAGE
- BKM_TTL_PAGES
- BKM_ENDPAGE

Filter: None

BKM_SEQ	BKM_INDENT	BKM_TEXT	BKM_START_...	BKM_TTL_PA...	BKM_ENDPAGE
1	1	Table 14.1.2.1.1..	1	5	1
2	1	Table 1 Number..	2	5	3
3	1	Table 14.1.6.1 ...	4	5	4
4	1	Figure 14.2.7.1 ..	5	5	5

– The BKM_START_PAGE variable is used to automatically create a hyperlink in the DEFINE.XML to the page in the bundled PDF


Thank you