SDTM Bookmarking Automation: Leveraging SAS, Ghostscript and Form-Visit Study Data

Nasser Al Ali, Genentech, South San Francisco, CA

Katrina Paz, Genentech, South San Francisco, CA

ABSTRACT

The United States Food and Drug Administration (FDA) requires that annotated Case Report Forms (aCRF) be submitted as part of the electronic data submission for every clinical trial. This annotated CRF is a PDF document that maps the captured data in a clinical trial to their corresponding variable names in the Study Data Tabulation Model (SDTM) datasets. The SDTM Metadata Submission Guidelines recommend that the aCRF should be bookmarked in a specific way. A one-to-one relationship between the bookmarks and aCRF forms is not typical; one form may have two or more bookmarks. Therefore, the number of bookmarks can easily reach thousands in any study! Generating the bookmarks manually is a tedious, time consuming job. This paper presents an approach to automate the entire bookmark generation process by using SAS® 9.2 and later releases, Ghostscript, a PDF editing tool, and leveraging the linkages between forms and their corresponding visits. This approach could potentially save tremendous amounts of time and the eyesight of programmers while reducing the potential for human error.

INTRODUCTION

Process automation is a key factor in achieving work efficiency, especially for time consuming and repetitive tasks that are prone to error. The manual generation of thousands of PDF bookmarks can certainly be considered one of these repetitious undertakings. Manual aCRF bookmark creation usually takes between several hours to a few days, depending on the number of forms and visits in the clinical trial. Thus, the automation of the aCRF bookmarking will not only reduce the processing time to a few minutes, but it will also prevent the possibility of human-related errors and inconsistencies across different studies.

The bookmark automation process discussed in this paper depends mainly on SAS® and relies on a number of existing documents, including the Case Report Forms (CRFs), the annotated Case Report Forms (aCRFs) and the Visit Form Matrix (VFM). For our purposes, the CRFs are generated by the RAVE Electronic Data Capture (EDC) system, but the concept and process may be applied to any set of CRFs that would be generated by any EDC system. The VFM, an Excel workbook that provides a link between CRF Forms and timepoints, is comprised of columns that are constituted by timepoints and rows that are constituted by Form names. If a form appears in any timepoint, the cell that corresponds to that timepoint/form will have a non-missing value. Display 1 shows an example of a VFM document. The standardization of CRFs and VFMs allows us to provide one solution that will work across all studies with minimum changes to the SAS® program's parameters.

In addition to SAS®, Ghostscript software, an interpreter for Postscript (PS) and PDF, is used in this process. PostScript is a programming language that was initially used to control the layout for printing pages. PostScript has an extension, pdfmark, which is used for generating and/or controlling PDF documents. In the automation process, PostScript is used to generate the bookmarks. Ghostscript is then used to merge the bookmarks with the aCRF PDF file.

This paper will focus on the bookmark automation process. The process steps are as the following:

1. Identification of input sources
2. Import of various data types into SAS® (definitions and techniques)
3. Preprocessing of data and preparation of an ordered listing to generate the bookmarks
4. Generation of the bookmark PostScript file and its merge with the aCRF PDF document

Joy and Couturier (2015) had previously covered the automation of the aCRF generation. They offered, at a high level, excellent ideas for the bookmark automation. This work is inspired by some of their ideas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	PRIMARY VISIT FORM MATRIX		X = Form always present, A = Form present once Visit Date form filled out.																
2	FOLDER OID			IVRS	MD	AE	PRC	SCRN	C1D1	C2D1	TTAW6	C3D1	C4D1	TTAW12	C5D1	C6D1	TTAW18	C7D1	C8D1
3	FOLDER NAME	Subject Level		IVRS	Concomitant Medications	Adverse Events	On-Study Procedures	Screening	Cycle 1 Day 1	Cycle 2 Day 1	Treatment Tumor Assessment Week 6	Cycle 3 Day 1	Cycle 4 Day 1	Treatment Tumor Assessment Week 12	Cycle 5 Day 1	Cycle 6 Day 1	Treatment Tumor Assessment Week 18	Cycle 7 Day 1	Cycle 8 Day 1
4	Folder sequence			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	Visit Calendar: Target			Not in	Not in	Not in	Not in	0	1	22	43	43	64	85	85	106	127	127	148
6	Visit Calendar: Overdue			Not in	Not in	Not in	Not in	7	7	9	7	9	9	7	9	9	7	9	9
7	Form Name	Form OID																	
8	Visit Date	VISIT						X	X	X	X	X	X	X	X	X	X	X	X
9	IVRS	IVRS1		X															
10	Subject Identification Note: P = Primary form. PTID form should always be marked P in the	PTID	P																
11	Subject Eligibility	ELIG						A											
12	Concomitant Medications Assessment	MDA1			X														
13	Concomitant Medications	MD1			MDA1														
14	Concomitant Medications - Coded	MD1C			X														
15	Adverse Event Assessment	AEDEA				X													
16	Adverse Events	AEDE				AEDEA													
17	Adverse Events - Coded	AEDEC				X													
18	Drug Safety Integration	DS				X													
19	SAE Reporting Summary	SAEREPORT				X													
20	AE Immune Mediated	DIAGP1				AEDEA													
21	AE Hypoxia	DIAGP2				AEDEA													
22	AE Pleural Effusion	DIAGP3				AEDEA													
23	AE Pericardial Effusion	DIAGP4				AEDEA													
24	On-Study Cancer Surgery Assessment	SRA2					X												

Display 1. Visit Form Matrix Excel Sheet

GATHERING INFORMATION

Three main components are needed to automate the bookmarking process using SAS® and Ghostscript: the bookmark's name, the corresponding CRF page number, and the associated number of sub-bookmarks. The main goal is to create a SAS® dataset with three variables; each variable contains data from one of the main components and each row corresponds to a bookmark. The dataset will then be used to write the bookmark as a PostScript file.

There are three links that will help in obtaining the three components mentioned above. These three mappings are visit-form name, form name-page number and domain name-form name. For the purposes of this paper, the form name refers to the CRF name and the page number specifies the CRF page number. All of this information may be gleaned from the VFM, CRF and aCRF documents.

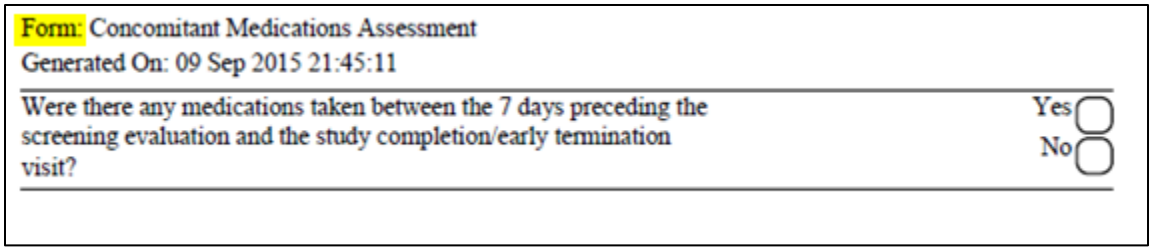
The link between the visits and form name, delineating which form is utilized in which visit, can be found in the VFM Excel workbook. The VFM structure allows us to use `proc import` to simply read the data into SAS®.

The mapping between form name and the corresponding page number can be found in the CRF PDF document. PDF files are mostly free text, and are composed of unstructured data, turning the extraction of information into a challenging task. The first step is to convert the PDF into a text file using the Ghostscript code below.

```
data _null_;
x "gs -dBATCH -dNOPAUSE -sDEVICE=txtwrite -sOutputFile=CRF.txt CRF.pdf";
run;
```

After running the Ghostscript code, we can use the SAS® `infile` statement in a data step to read the text file line by line. The CRF's design and format allows us to find a unique pattern to identify the form name and page number. This pattern recognition is exemplified in Display 2, a sample of a CRF page. Notably, the form name always comes after the word "Form:"; hence, an `if`-statement and `substr` function can be used to extract the form's name. Moreover, this pattern appears once per page and the page number will increment whenever this pattern occurs.

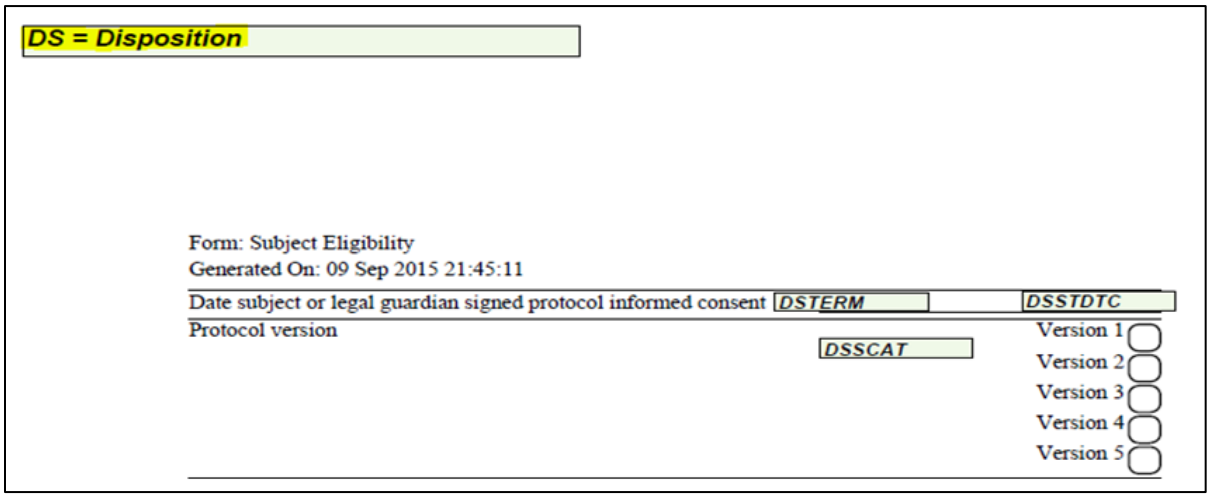
```
if (index(_infile_,"Form:") GT 0) then do;
                  pagen+1;
                  form =
trim(left(substr(_infile_,index(_infile_,"Form:")+6)));
    end;
```



Display 2. Case Report Form

The previous technique that involves finding a unique pattern and extracting the information accordingly can be applied to obtain the link between domain name and form name. Each form in the aCRF might contain one or more domain names. The domain names appear at the top of the page and can be identified by two capital letters followed by an equal sign. For example, the adverse event domain would be marked as "AE = Adverse Event." For this step, we can use SAS® `prxmatch` function as shown in the code below. This process will result in a list that allows us to establish which form appears in which domain.

```
ptrn = "/[A-Z][A-Z] =/";
if (prxmatch(ptrn, _infile_) GT 0) then
     fname = trim(left(substr(_infile_,index(_infile_,"=")+1)));
```



Display 3. Annotated Case Report Form

BOOKMARKS STRUCTURE

The SDTM Metadata Submission Guidelines recommend that the aCRF should be bookmarked in two ways: by timepoints such as planned visits and by domain.

- Bookmark by timepoint is ordered chronologically according to the Time and Event (T&E) schedule. Within each timepoint, form names should appear according to their page number in the aCRF.
- Bookmark by domain is ordered alphabetically. Under each domain, timepoints are ordered according to the T&E schedule. Additionally, within each timepoint, form names are ordered by page number.

BOOKMARK BY TIMEPOINT

In the previous section, the importing of timepoints-form names and form names-page number links from the VFM and CRF documents were discussed. The SAS® `transpose` procedure can be used to create a single column of all timepoints since each timepoint in the VFM is stored in a column. The VFM also has a sequence number assigned to each timepoint according to the T&E schedule. This allows the program to order the timepoints properly, creating a listing by looping through each timepoint and keeping only forms that appear in the current timepoint. This will result in a single column with all timepoints, and under each timepoint its correspondent forms will be displayed (second column in Display 4).

The created column can now be merged with the form name-page number data that was obtained from the CRF. Now that the page number has been derived, a new variable named *order* can be created for sequencing purposes. This variable will have a value of "0" if the row corresponds to a timepoint; otherwise, it will have a value equal to the form's page number. The creation of the *order* variable allows the data to be sorted first by the variable folderseq and then by the page order.

In addition, a new variable, named *child,* is created to count the number of sub-bookmarks under each timepoint. A simple `proc sql` can achieve this task. The final dataset listing can be seen in Display 4.

Obs	ttle	pagen	foldersq	order	child
1	Screening	1	5	0	39
2	Visit Date	2	5	2	0
3	Subject Eligibility	3	5	3	0
4	Pre-Screening Tumor Tissue Research Sample Informed Consent	26	5	26	0
5	Optional Fresh Tumor Biopsies Research Sample Informed Consent	28	5	28	0
6	Optional Leftover Tumor Tissue Research Sample Informed Consent	30	5	30	0
7	Demographics	34	5	34	0

Display 4. Timepoint Bookmark Listing

BOOKMARK BY DOMAIN

The Domain bookmarking is relatively easy since most of the work has already been done. The first step is to get a list of all domains ordered alphabetically. Following that, the program will loop through each domain, and then through each visit; the domain name-form name link, obtained from the aCRF, is used to keep forms that appear in the current domain and visit. The timepoints and forms under each domain will be ordered similarly to the sequencing utilized in the timepoint bookmarking. The variable, child, that holds the count of sub-bookmarking under each domain/timepoint is also created here.

PUTTING THINGS TOGETHER

The final step involves combining the two generated listings, timepoints and domain, using a simple `set` statement in a data step. Prior to this, the first level of the bookmarks with the study identifier needs to be created. The study identifier, as seen in the first row of Display 5, will have two sub-bookmarks: visits and domains. Display 5 shows the final ordered dataset that is ready to be converted into a PostScript file.

Obs	ttle	pagen	foldersq	order	child
1	**Study Identifier**	1	.	.	2
2	Visit	1	.	.	124
3	Screening	1	5	0	39
4	Visit Date	2	5	2	0
5	Subject Eligibility	3	5	3	0
6	Pre-Screening Tumor Tissue Research Sample Informed Consent	26	5	26	0
7	Optional Fresh Tumor Biopsies Research Sample Informed Consent	28	5	28	0
8	Optional Leftover Tumor Tissue Research Sample Informed Consent	30	5	30	0
9	Demographics	34	5	34	0

Display 5. Final Bookmark Dataset Listing

CREATING THE BOOKMARK POSTSCRIPT FILE

Once the pdfmark syntax is known, it is easy to create the PostScript file using a data step with the bookmark dataset created previously. The dataset has three variables: title, the bookmark's text; pagen, the bookmark's page number; and child, the number of sub-bookmarks. A PS file called bookmark is first generated then a `data _null_` procedure with a `put` command writes to the file. The PS file starts with '%!PS-Adobe-3.0' and ends with '%%EOF'.

When generating a bookmark using PS code, each bookmark syntax starts with "[/Parameter1" and ends with "/OUT pdfmark". There are two required parameters: Title, the bookmark's text; and Count, the number of sub-bookmarks under each bookmark. Notably, a negative sign is used with the Count option so that the bookmarks are collapsed when the PDF file is opened. Moreover, there are several optional parameters, two of which are used here: Page, the bookmark's page number; and View, the adjustment for the page's view if the bookmark is clicked on. The /FitB option is used so that the PDF page will fit the window border. The code below is used to generate the PostScript file.

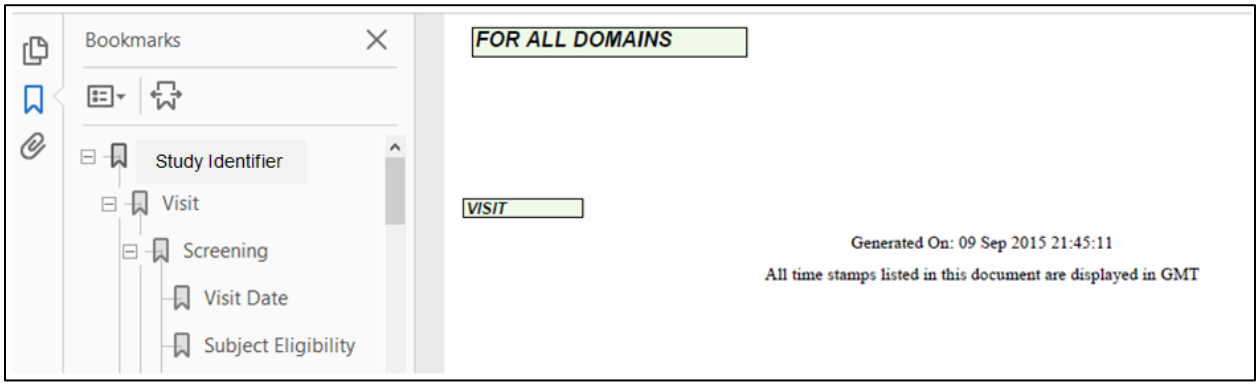
```
filename bookmark "bookmark.PS";
data _null_;
    set final end = eof ;
    file bookmark ls=10000;
if (_n_ eq 1) then put '%!PS-Adobe-3.0' /;

 attrib ttle1 length = $800;
          x = '(';
          y = ')';
    call cats(ttle1,x,ttle,y);
 put
    '[/Count -' child
    / '/Page ' pagen
    / '/View [/FitB ]'
    / '/Title ' ttle1
    / '/OUT pdfmark';
if (eof) then put  / '%%EOF';
run;
```

COMBINING POSTSCRIPT WITH ACRF

The last step to complete bookmarking involves combining the bookmark PostScript file with the aCRF PDF file using Ghostscript software. The code for this step is shown below.

```
DATA _NULL_;
x "gs -q -dBATCH -dNOPAUSE -sDEVICE=pdfwrite -sOutputFile=aCRF.pdf bookmark.PS
no_bookmark_aCRF.pdf";
Run;
```



Display 6. Final aCRF PDF file after adding the bookmark.

CONCLUSION

This paper demonstrates a way to programmatically bookmark the annotated Case Report Form (aCRF) with Study Data Tabulation Model (SDTM) variables for submission to the FDA. This approach encompasses bookmarking by both timepoint and domain. Notably, the process requires SAS® and Ghostscript software and the availability of Case Report Forms, annotated Case Report Forms and the Visit Form Matrix. While this approach is best undertaken by advanced programmers, the time saved and the decreased risk for manual errors justifies this option as favorable for submissions to the FDA.

REFERENCES

- Joy, Geo, and Couturier, Andre. (2015). “SDTM Annotations: Automation by implementing a standard process.” Proceedings of the PharmaSUG 2016 Conference. Available at http://pharmasug.org/proceedings/2015/AD/PharmaSUG-2015-AD07.pdf
- PDFMARK Reference Manual. Available at http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdfmark_reference.pdf

- Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG), prepared by the CDISC SDS Metadata Team. Available at http://www.cdisc.org/system/files/members/standard/application/zip/final_metadata.zip
- U.S. Food and Drug Administration, “Study Data Technical Conformance Guide v3.0.” March 2016. Available at http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf

ACKNOWLEDGEMENTS

We would like to thank our colleagues Amy Klopman, Dyuthi Yellamraju, Nelson Lee and Michael Brodie for their technical support and editing assistance.

CONTACT INFORMATION

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

Name: Nasser Al-Ali
Enterprise: Genentech Inc, A Member of Roche Group
Address: 1 DNA Way
City, State ZIP: South San Francisco, CA 94080
Work Phone: +1 (415) 316-8646
E-mail: alalin@gene.com
Web: https://www.gene.com

Name: Katrina Paz
Enterprise: Genentech Inc, A Member of Roche Group
Address: 1 DNA Way
City, State ZIP: South San Francisco, CA 94080
Work Phone: +1 (650) 826-9914
E-mail: paz.katrina@gene.com
Web: https://www.gene.com