Automatic generating blankcrf.pdf for Rave Study

Haiqiang Luo, PPD Inc., Beijing
Yong Cao, PPD Inc., Beijing

ABSTRACT

blankcrf.pdf is a critical component of eCRT for NDA submission. It is a blank CRF with annotations that documents the location and mapping of CRF data with corresponding data set names and variables names in the tabulation data set. Manual creating blankcrf.pdf is a common practice. However, it is a tedious repetitive task which can lead to inconsistence issue among sibling studies. The standardized structure of CRF produced by Medidata Rave System makes it feasible to automate the mapping process. This paper describes a method of generating blankcrf.pdf automatically for this kind of studies.

INTRODUCTION

The blankcrf.pdf shows the mapping between the data from CRF and the data in tabulation data set as straightforward. The file should be prepared at the beginning of SDTM work, and as guidance for SDTM specification writing. In practice annotated blankcrf.pdf is usually created manually project by project. The process is time-consuming and may increase the risk of inconsistence in a series of sibling projects.

For the study which used Medidata Rave system, the CRF book is built in a special structure. Each form is named by the raw data set name. Then the forms are grouped and ordered by Folders (Visit) in the bookmark. On CRF page, each field has a link to variable metadata page. Rave provides a SDS (Study Design Specification) file where fields are listed in the same order as they appeared on the CRF pages. Adobe Acrobat® is an industry standard tool for advance processing of PDF files. This paper utilizes Acrobat embedded JavaScript to extract bookmarks and links from Rave CRF and then uses some SAS® macros to create a Forms Data Format (FDF) file which contains all annotations and can then be imported into the blank CRF to get the final blankcrf.pdf.

ADOBE ACROBAT FDF FILE

To define an annotation text box in CRF, the follow information is mandatory,
1. The properties of the annotation (rectangle) box:
   1) Fill color
   2) CRF page and location
   3) The width and height of the box

2. The properties of the annotation text:
   1) Font and size, Italic/Bold, Alignment and Color
   2) Text

Figure 1 shows an example of an FDF file exported by Adobe Acrobat(Choose "Comments">>"Export Comments to Data File…"). The example contains single annotation box of RACE. All the parameters above can be found in the file.

The FDF file is made up of 3 types of objects: header, body and trailer. The header object appears at the top of the FDF file and describes the number of annotations as well as the annotation source file name. The header object followed describes each annotation's attributes. The trailer object is at the bottom of the FDF file signaling the end-of-file.

```
%FDF-1.2
%âãÏÓ
1 0 obj
<</FDF<</Annots[2 0 R]/F(/K/SDTM/_SDTM
practice/acrf.pdf)/ID[<F06D83C43F71CF4B9BF15296D2241EE0><C7264C3426091C418D9C82AFD6A79734>]/UF(/K/SDTM/_
SDTM practice/acrf.pdf)>>/Type/Catalog>>
endobj
```

```
2 0 obj
<</C[0.75 1.0 1.0]/Contents(RACE)/CreationDate(D:00000000000000Z)/DA(/Helv 10 Tf)/DS(font: italic bold Helvetica 10.0pt;
text-align:center; color:#FF0000 )/NM(fh7N6rDk6wVStcbR9vhDiA)/Page 1/RC(<?xml version="1.0"?><body
xmlns="http://www.w3.org/1999/xhtml" xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/" xfa:APIVersion="Acrobat:9.0.0"
xfa:spec="2.0.2"   style="font-size:10.0pt;text-align:center;color:#FF0000;font-weight:bold;font-style:italic\
;font-family:Helvetica;font-stretch:normal"><p>RACE</p></body>)/Rect[426.904 398.0 471.704 415.0]/Subj(Text
Box)/Subtype/FreeText/T(luoh)/Type/Annot>>
endobj
trailer
<</Root 1 0 R>>
%%EOF
```

Figure 1: Acrobat FDF file example

Body object of RACE can be simplified to keep only required attributes, see Figure 2.

```
2 0 obj
<</C[0.75 1.0 1.0]/Contents(RACE)/DS(font: italic bold Helvetica 10.0pt; text-align:center; color:#FF0000 )/Page 1/Rect[426.904
398.0 471.704 415.0]/Subtype/FreeText/Type/Annot>>
endobj
```

Figure 2: Simplified attributers for body object

DEVELOP TEH SAS MACRO

Upon understanding the structure of FDF file, it is intuitive to create it using SAS ®:
1. Determine the properties of annotation (rectangle) box:
   1) Fill color: Use default values(setup as Latex color: http://latexcolor.com/). Need to color each of the domains a different color when more than one domains in the same CRF page.
   2) CRF page and location: Derive from bookmarks file and links file
   3) The width and height of the box: Can be determined by two coordinates of points in diagonal. It also will be based on the annotation text (see next section).

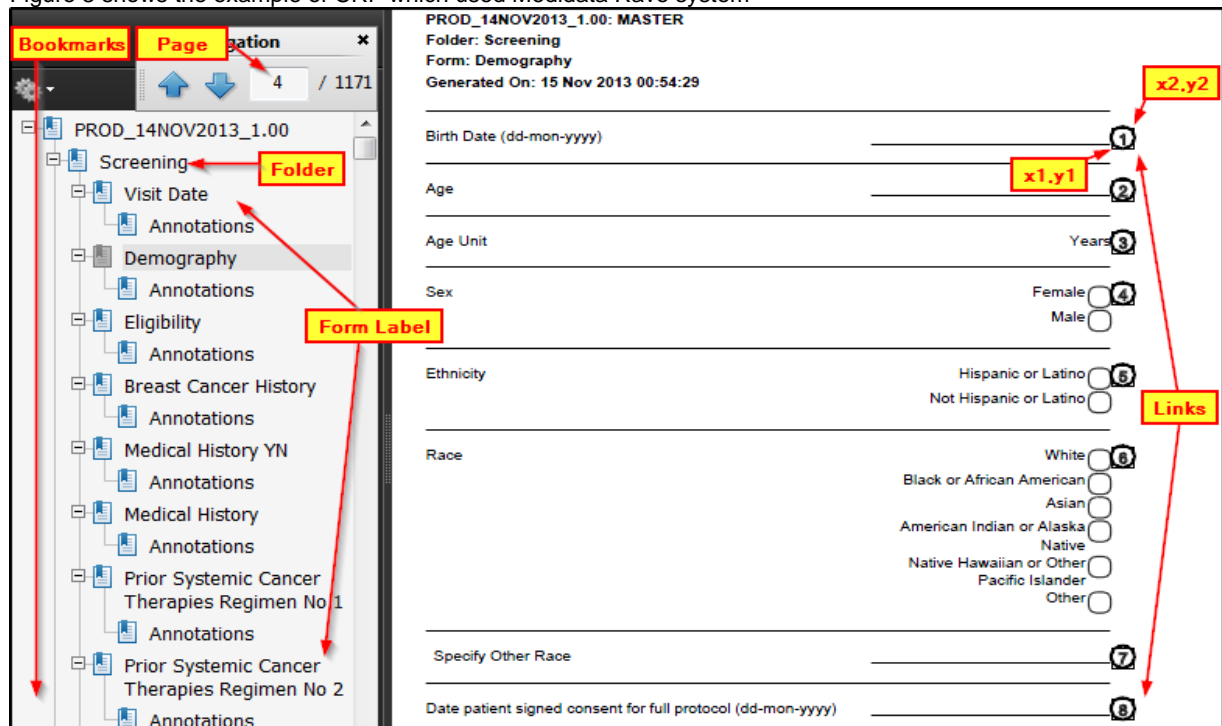Figure 3 shows the example of CRF which used Medidata Rave system

Figure 3: Rave CRF example

Bookmarks and Links along with their attributes in Rave CRF Form can be exported using JavaScript (**See Appendix**). JavaScript is a cross-platform scripting language that is widely used in Web browser. It is also included in Adobe Acrobat Product Family. For more details, refers to http://www.adobe.com/cn/devnet/acrobat/javascript.html.

Figure 4 is a screenshot of exported bookmarks file and figure 5 is a screenshot of exported links file.

Figure 4: Bookmarks file	Figure 5: Links file

Form label and its page number are listed in the bookmarks file.
From the links file, we can know page number and coordinate (as show in figure 3) of each link. Since those links are equivalents of form fields and they are listed in the order as they appeared on the Form, we can use this file along with Rave SDS file to locate any specific field.

2. Determine the properties of annotation text:
    1) Font and size, Italic/Bold, Alignment and Color: Use default values:
        Domain Text: italic bold Helvetica 12.0pt; text-align:center; color:#FF0000
        Variable Text: italic bold Helvetica 10.0pt; text-align:center; color:#FF0000
    2) Text in annotation:
        As a core concept of this paper, we use blankcrf Definition Spreadsheet to automate the mapping from Form Field to SDTM variable. The blankcrf Definition Spreadsheet is a dictionary of mapping rules, see figure 6.
        The basic assumption is when Field Name or CRF Question Text of a study match with the value in column E-F in the spreadsheet, then will create annotation text based on column A-D. Such as:
        ● "BRTHDTC" will be annotated when CRF question is "Birth Date" or "Date of Birth".
        ● "DSSTDTC where DSTERM = "INFORMED CONSENT OBTAINED"" will be provided when CRF question text contains "Date" and "Consent"
        ● "VSORRES where VSTESTCD = WEIGHT" will be provided when the field name is "WEIGHT".
        ● "SUPPDM.QVAL where QNAM = RACEOTH" will be provided when the field name is "RACEOTH" or the question text contains "Other Race".

We need to develop codes for each type of mapping rule.

	A	B	C	D	E	F	G
1	Domain	SUPP	TopicValue	Variable	Field Name	CRF Question Text	Implementation Notes:
2	DM			BRTHDTC	BRTHDAT	*Birth Date*|*Date of Birth*	1) In Field Name and CRF Question
3	DM			AGE	AGE	Age|Age (*	Text column:
4	DM			AGEU	AGEU	Age Unit	* means any characters.
5	DM			SEX	SEX	Sex|Gender		means "or"
6	DM			ETHNIC	ETHNIC	Ethnicity	
7	DM			RACE	RACE	Race	2) When match with Field name or CRF
8	DM			COUNTRY	COUNTRY	Country	Question Text(ignore the case of the
9	DM	Y	RACEOTH		RACEOTH	*Other Race*	characters), then will create
10	DS		DSSTDTC where DSTERM = "INFORMED CONSENT OBTAINED"	DSTERM		*Date*Consent|*Consent*Date	annotations based on column A-D
11	DS		DSSTDTC where DSTERM = "RANDOMIZED"	DSTERM	RANDAT	*Date of Randomization*	
12	VS		SYSBP	VSTESTCD	SUPSYS		
13	VS		DIABP	VSTESTCD	SUPDIA		
14	VS		HEIGHT	VSTESTCD	HEIGHT		
15	VS		WEIGHT	VSTESTCD	WEIGHT		
16	VS		VSALL	VSTESTCD	VSPERF	Vital Signs performed*	
17	AE		Not Submitted		AEYN		
18	AE			AETERM	AETERM	Adverse Event	
19	AE			AESEV	AESEV		
20	AE			AESER	AESER	Serious	
21	AE			AESTDTC	AESTDAT	*AE* Start Date*|*Adverse Events* Start Date*	

Figure 6: blankcrf Definition Spreadsheet example

The best practice is to use this blankcrf Definition Spreadsheet as an central repository of mapping rules across many projects, rather than maintaining different ones for different projects.

The SAS macro parameter can be defined as below:

```
%macro sCRF_Auto_RAVE(
/*** inputs ***/
        bookmark_file       =
,       CRF_link_file       =
,       sds_file            =
,       blankcrf_def_file   =
/*** outputs ***/
,       FDF_file            =_smartFDF
/*** default setting ***/
,       C_Property0 =0.75 1.0 1.0             /* Latex color: light green */
,       C_Property1 =1.0 1.0 0.666656         /* Latex color: light yellow */
,       C_Property2 =1.0 0.75 0.666656        /* Latex color: light orange */
,       C_Property3 =1.0 0.666656 0.75        /* Latex color: light pink */
,       C_Property9 =0.75 0.75 0.75           /* Latex color: light purple */
,       Prop_Domain =%str(italic bold Helvetica 12.0pt; text-align:center; color:#FF0000)
,       Prop_Var    =%str(italic bold Helvetica 10.0pt; text-align:center; color:#FF0000)
,       unit=6.2                              /* width of 'a' in Rave CRF */
,       offsety=17                            /* height of annotation for variables */
);
```

Then the macro call will become very simple:

```
%sCRF_Auto_RAVE(bookmark_file        =acrf.txt
,               CRF_link_file        =acrf.pdf.links.txt
,               sds_file             =U:\A Project\Pharma 1\Project1_PROD_14NOV2013.xlsx
,               blankcrf_def_file    =U:\CDISC\_SDTM practice\blankcrf definition.xlsx);
```

Figure 7 is the final blankcrf.pdf, which is the source CRF with imported FDF file(Choose "Comments">>"Import Comments…") that be generated by SAS macro above, without any further manual action.
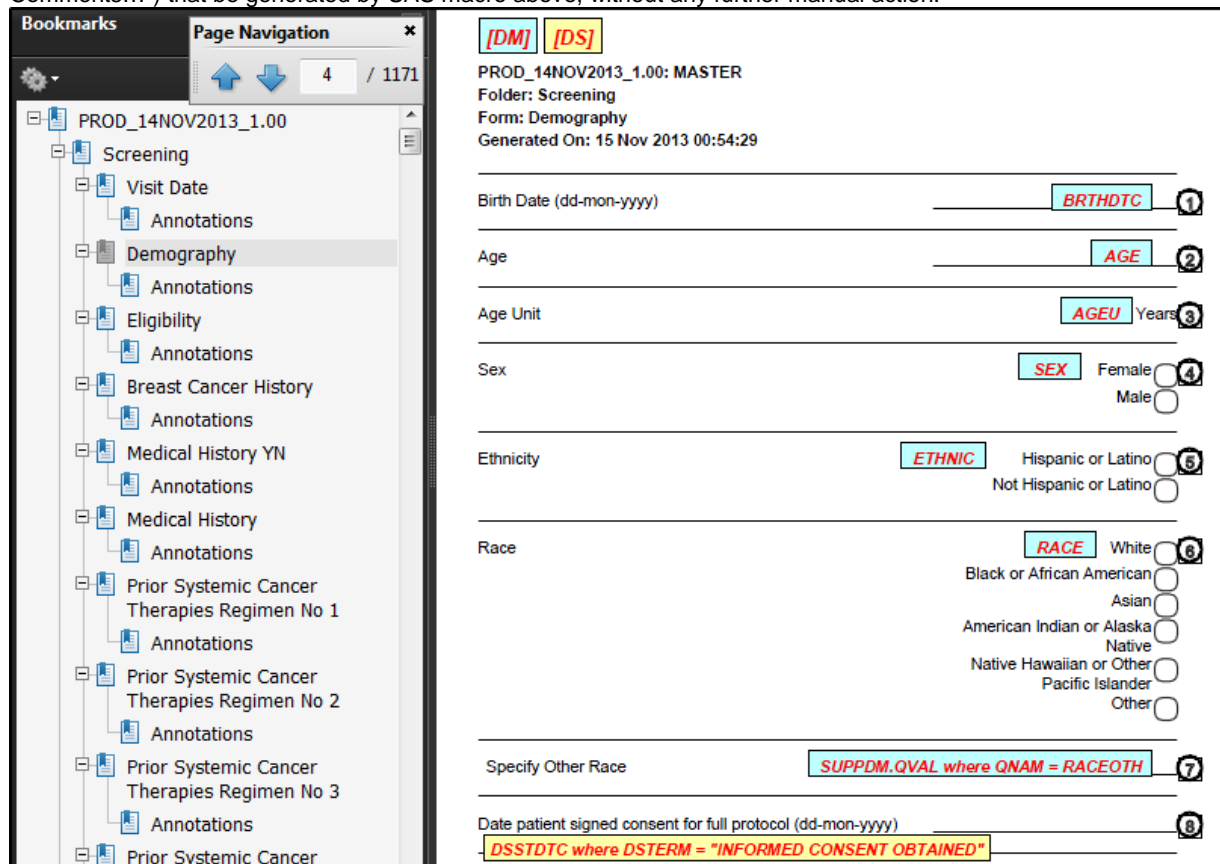

Figure 7: Final blankcrf.pdf

Automatic generating blankcrf.pdf for Rave Study, Continued

CONCLUSION

Preparing blankcrf.pdf is a tedious, repetitive task. However, the special structure of CRF book in Medidata Rave system makes it possible to automate the process. As demonstrated in this paper, SAS can be used to create annotations file (Adobe FDF file) based on mapping rules setup in blankcrf Definition Spreadsheet. The benefit can grow when more mapping rules are added to the Definition Spreadsheet.

REFERENCES

Automating Production of the blankcrf.pdf, Walter Hufford, Novartis, East Hanover, NJ

Have SAS Annotate your Blank CRF for you! Plus dynamically add color and style to your annotations., Steven Black, Agility-Clinical Inc., Carlsbad, CA

ACKNOWLEDGMENTS

I would like to thanks Sherry Zhao (Sanofi R&D) for her review and suggestions.

CONTACT INFORMATION

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

> Name: Haiqiang Luo
> Enterprise: PPD Inc.
> 25 F Raffles City Beijing Office Tower, No. 1 Dongzhimen South Street. Dongcheng District
> Beijing, 100007
> Work Phone: +86 10 61846097
> Fax: +86 10 61846099
> E-mail: Haiqiang.Luo@ppdi.com
> Web: www.ppdi.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Adobe, the Adobe logo, Acrobat and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
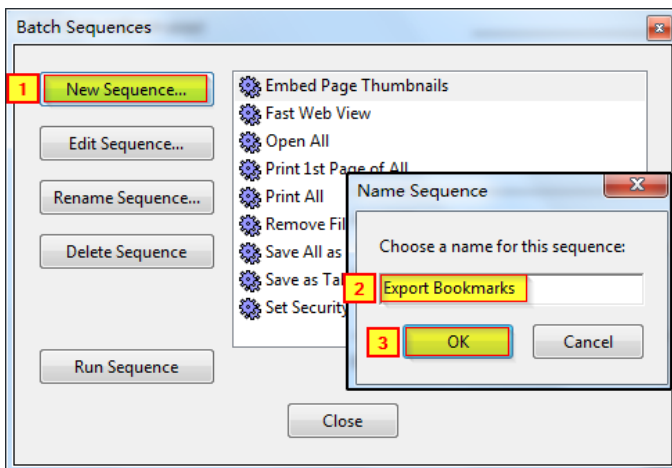
APPENDIX:
The instruction below will illustrate on how to export bookmarks and links using Acrobat JavaScript. The screenshots are based on Adobe Acrobat 9.

- Goto "Advance">>"Document Processing">>"Batch Processing…"
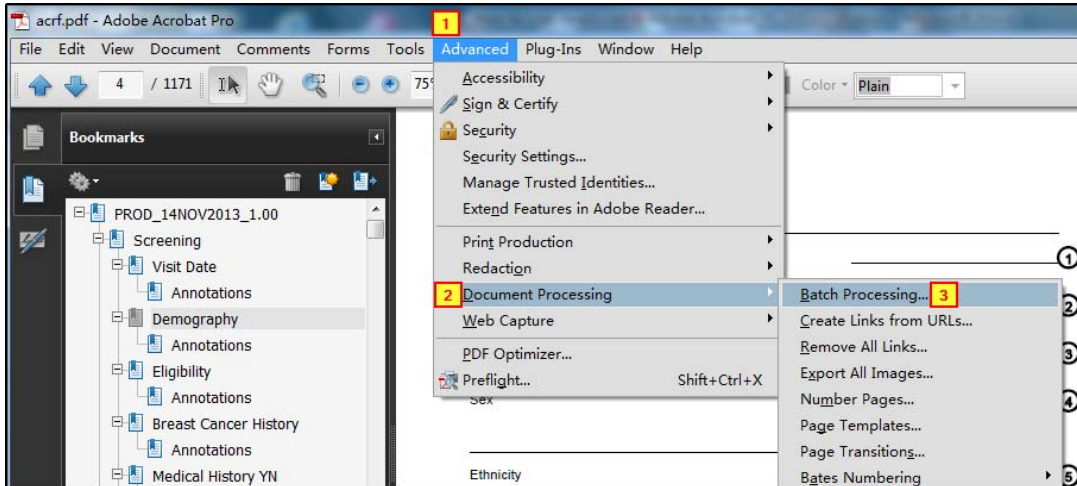
- Select "New Sequence" in the list on the left of the panel.

- Enter "Export Bookmarks" for the name of the sequence.

- Select "OK" to save the Sequence name.

- Select "Select Commands"

- In "Edit Sequence" window, locate to "JavaScript" category, select "Execute JavaScript" and then select "Add>>"

- Select "Edit", then copy and paste below scripts to the "JavaScript Editor" window.

- Select "OK" to save the script.
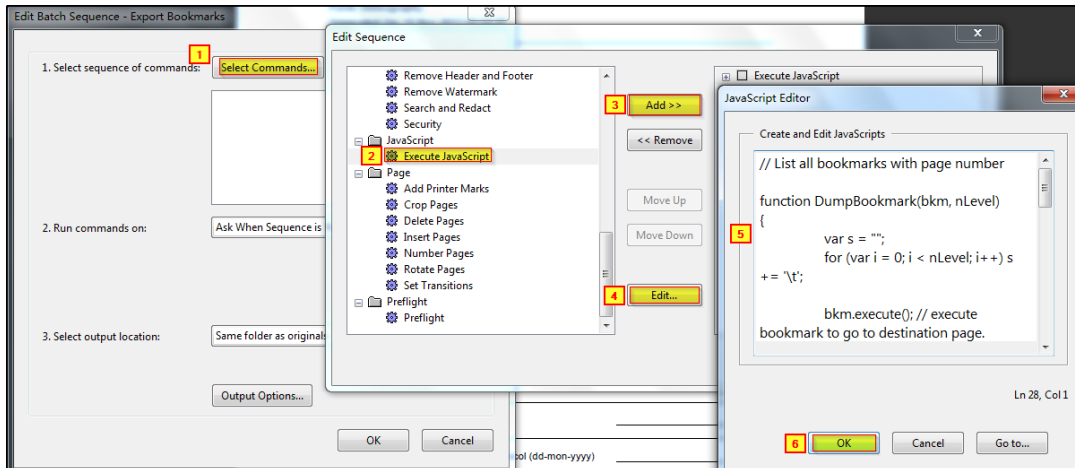
```
function DumpBookmark(bkm, nLevel)
{
var s ="";
for(var i =0; i < nLevel; i++) s +='\t';
    bkm.execute();// execute bookmark to go to destination page.
var pageN =this.pageNum +1;// internally in PDF page # start with 0
if(bkm !=this.bookmarkRoot)
    bmExp = bmExp + s + bkm.name +'\t'+ pageN +"\r\n";
if(bkm.children != null)
for(var i =0; i < bkm.children.length; i++)
    DumpBookmark(bkm.children[i], nLevel +1);
}

// console.clear(); console.show();
// console.println("Dumping all bookmarks in the document.");
var currDir =this.path.substring(0,this.path.length-this.documentFileName.length-1);
var fileName =this.documentFileName.substring(0,this.documentFileName.length-4);
var bmExp ="";
DumpBookmark(this.bookmarkRoot,-1);
this.createDataObject({cName:fileName +".exportType=Bookmark.txt", cValue: bmExp});
```

- Select "Export Bookmarks" and then select "Run Sequence"
- The exported bookmarks file will be attached to the PDF file
- Select "Attachment" in the bottom left corner.
- Right click on the file and select "Open Attachment" or "Save Attachment…" to open/save the attached file.
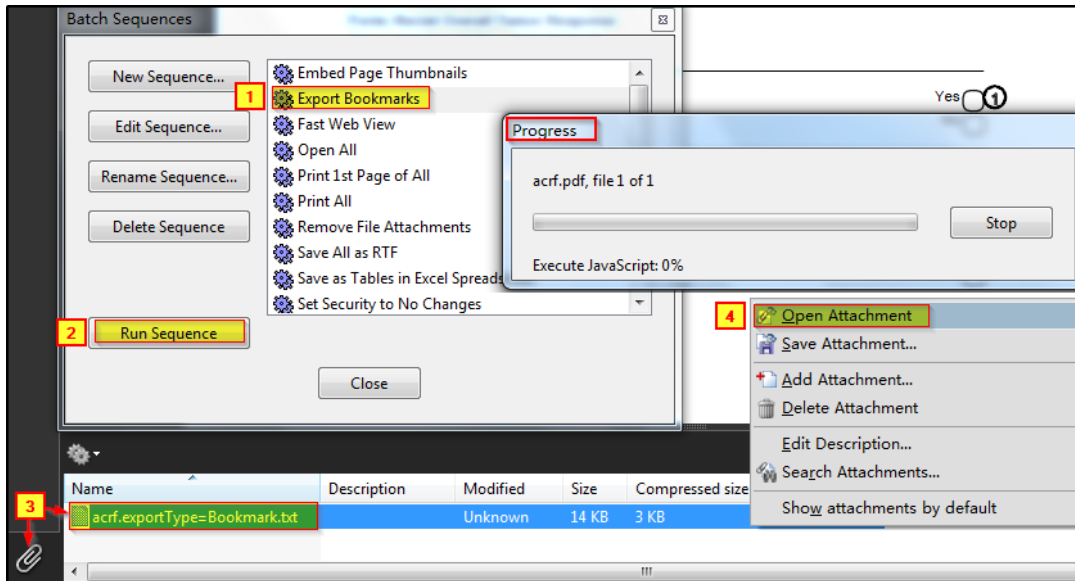
- Follow steps above to add "Export Links" batch sequence (using scripts below) and export the links.

```
// console.clear(); console.show();
// console.println("Dumping all links in the document.");
var currDir =this.path.substring(0,this.path.length-this.documentFileName.length-1);
var fileName =this.documentFileName.substring(0,this.documentFileName.length-4);
var numLinks =0;
var linkExp ="";

for(var pageN =0; pageN <this.numPages; pageN++)
{
var pBox =this.getPageBox("Crop", pageN);
var links =this.getLinks(pageN, pBox);
var pageNOffset = pageN +1;
for(var linkN =0; linkN < links.length; linkN++)
{
var borderStyle ="S";// Style of link border
var linkVisual =0;// Visibility of link border

// convert border color RGB vector to a single integer
var borderColor = links[linkN].borderColor[1]+
        links[linkN].borderColor[2]*Math.pow(2,8)+
links[linkN].borderColor[3]*Math.pow(2,16);
        linkExp = linkExp
+ pageNOffset +","
+ links[linkN].highlightMode.substring(0,1)+","
+ borderStyle +","
+ links[linkN].borderWidth +","

+ linkVisual +","

+ borderColor +","

+ links[linkN].rect[0]+","

+ links[linkN].rect[2]+","

+ links[linkN].rect[1]+","

+ links[linkN].rect[3]

+"\r\n";

}

}

this.createDataObject({cName:fileName +".exportType=Links.txt", cValue: linkExp});
```