

Automating Production of the blankcrf.pdf in a CRO Environment

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

ABSTRACT

The annotated CRF (blankcrf.pdf) is an important part of a submission. Per the CDISC SDTM Metadata Submission Guidelines the annotated CRF should be annotated with all domains and variables, and where necessary distinguish between multiple domains on the same CRF page, e.g. by colour coding domains. In addition the annotated CRF should include dual bookmarking (by timepoint and CRF topic) to allow easier navigation. Working in a CRO environment it is often not possible to have any control over the design of original CRF/eCRF forms as different sponsors may maintain their own global libraries of form designs. This paper describes an easy to use method of extending Adobe® Acrobat® using JavaScript™ code to ease the creation of the annotated CRF.

INTRODUCTION

Creating the blankcrf.pdf can be a laborious manual process that can take a long time, and be prone to errors. Considering the number of pages in a typical CRF and the number of fields that need to be annotated, the number of annotations quickly rises into the hundreds. Creating each annotation is a time consuming process, made more complex when you want to change one or more of font size, font colour, or annotation fill colour. In addition it is then often necessary to re-size the annotation to fit the text content. This paper introduces a method to simplify the creation of the blankcrf.pdf by scripting the creation of the annotations, applying changes to font size, colour and fill colour. A final step of re-positioning the annotations can then be done manually to ensure that the original CRF content is not hidden by annotations. To complete the creation of the blankcrf.pdf it is necessary to add dual bookmarks to aid navigation. Again, this is a time consuming process that can be automated using JavaScript.

CDISC SDTM METADATA SUBMISSION GUIDELINES

CDISC made available the Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG) in late 2011. The document provides guidance on how to compile the eCTD module 5 “sdm” folder, including the annotated CRF, submission datasets, and define.xml file. This paper will concentrate on the creation of the annotated CRF file, with suggestions on how to more easily create the annotations and bookmarks.

Preparation of the CRF to be annotated and bookmarked is outside the scope of this paper. It is assumed that the starting point is a PDF version of the complete CRF, with each unique form only present once in the PDF.

ANNOTATION GUIDANCE

The SDTM-MSG provides guidance on what to annotate along with suggestions for the text to use, in compliance with FDA Study Data Specifications. It is assumed that the reader is aware of what to annotate; instead, this paper will concentrate on the optional recommendations that the guidance provides to aid the reviewer in distinguishing between variable names, and variables within domains: (see Figure 1)

- To distinguish the domain level annotations from the variable annotations a slightly larger font, in a different colour can be used for the domain annotations.
- If more than one SDTM domain exists on a page then each domain annotation, and all of its variables, can be colour-coded.

PhUSE 2015

DM=Demographics		Screening
CDISC Study: CDISC01 STUDYID		Assessment Date SCDTC / /
DEMOGRAPHY		
Date of Birth: / / BRTHDTC		
SEX		
Gender: <input type="checkbox"/> Male <input type="checkbox"/> Female		
ETHNIC		
Ethnicity: <input type="checkbox"/> Hispanic or Latino <input type="checkbox"/> Not Hispanic or Latino		
RACE		
Race: Check all that apply		
<input type="checkbox"/> White	RACE , when more than one selected, RACE=MULTIPLE and individual responses are RACE1, RACE2, etc. in SUPPDM	
<input type="checkbox"/> American Indian or Alaska Native		
<input type="checkbox"/> Black or African American		
<input type="checkbox"/> Native Hawaiian or Other Pacific Islander		
<input type="checkbox"/> Asian		
<input type="checkbox"/> Other: RACEOTH in SUPPDM		
SC=Subject Characteristics		
FamilyStatus: <input type="checkbox"/> Never Married <input type="checkbox"/> Domestic Partner		
<input type="checkbox"/> Married	SCORES when SCTESTCD = MARISTAT	
<input type="checkbox"/> Legally Separated		
<input type="checkbox"/> Widowed		

Figure 1: Suggested formatting of annotations in SDTM-MSG

BOOKMARKING GUIDANCE

The annotated CRF should be bookmarked in 2 ways (dual bookmarking): (see Figure 2)

- By time point**
 This can be compared to the study visit schedule, although an additional bookmark will often be created to group log style pages, e.g. Adverse Events, Concomitant Medications. Inside each “time point” bookmark, a sub-level of bookmarks is created to list all CRF unique forms in that time point.
- By CRF topic**
 This can be compared to the title of the unique CRF forms. Inside each “CRF form” bookmark, a sub-level of bookmarks is created to list all time points the CRF is used.

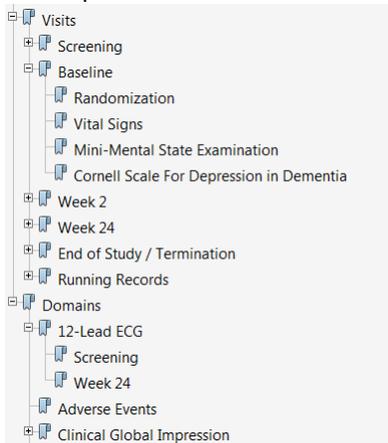


Figure 2: Suggested dual bookmarking in SDTM-MSG

EXTENDING ADOBE ACROBAT WITH JAVASCRIPT

Extending Microsoft® Word® or Excel® is common place using VBA Macros. In a similar way it is possible to extend the functionality of Adobe Acrobat using JavaScript code. (This does not apply to the Adobe Acrobat Reader product.) As part of the installation of Adobe Acrobat, a folder called “Javascrpts” is created. (The location of this folder will vary depending on the version of Windows in use, and the version of Adobe Acrobat installed.) By saving JavaScript files to this location it is possible to customize how Adobe Acrobat functions, or to automate what would otherwise be a long manual process.

Adobe Acrobat has a comprehensive API that has been developed and extended as the product has developed over its life.

PhUSE 2015

CREATING ANNOTATIONS

Using the example from the SDTM-MSG we can identify the requirements for the annotations that will be created by JavaScript as:

- different font size
- different font colour
- different fill colour
- fix size of annotation to match annotation text

After reviewing the API, the `addAnnot` method can be used. This takes a single object parameter that in turn specifies the various attributes that describe the annotation appearance. After creation of the annotation a second call is made to set the text contents of the annotation. The following code shows how a single annotation can be created. In order to calculate the size of the annotation box a monospaced font has been used, in this case Courier. If the reader wishes to use a different font, it is suggested that a monospaced font be used to make calculations of the annotation width easier. Using a font that is not monospaced is certainly possible, but it is probable that the reader would then need to re-size each annotation as a manual step.

```
// depending on annotation type - dataset/variable - different font size/colour
if (_type == "D") {
    fsz = 12;
    fgc = color.black;
} else {
    fsz = 8;
    fgc = color.red;
}

// determine position (hard-coded) and size of bounding box (dependant on text
length)
x1 = 100;
y1 = 400;
x2 = x1 + (_text.length+1)*fsz*0.6;
y2 = y1 + fsz + 2;

// lookup RGB colour definition
bgc = acrf_color(_colour);

// construct the array for the text with formatting
rt = new Array();
rt[0] = new Object();
rt[0].text = _text;
rt[0].textColor = fgc;
rt[0].textSize = fsz;
rt[0].fontFamily = ["Courier", "monospace"];

// create the annotation
annot = obj.addAnnot({
    page: _page-1,
    type: "FreeText",
    rect: [x1, y1, x2, y2],
    width: 1,
    strokeColor: color.black,
    fillColor: bgc,
    alignment: 1
});
annot.richContents = rt;
```

The JavaScript code presented above only allows for a single line of text in the annotation that is created. In the author's experience there are very few occasions where a multiple line annotation may be required, see Figure 1 for one such example where logic for the RACE variable is included in the annotation. For any such annotations the reader would manually resize the annotation at the same time as positioning them on the page. Updating the code to allow for multi-line annotations is left as an exercise for the reader; `x2` and `y2` would need to be updated to account for multiple lines of text, and the `rt` array would need an entry for each line of text.

PhUSE 2015

Working in a CRO environment and being exposed to CRFs designed by multiple sponsors the position of each annotation on the page is not known. In contrast, if you have control over the design of the CRF you may know the intended annotation location on the page. If this is the case, variables x1 and y1 in the above code could be updated.

The above code is contained in a JavaScript function that can then be called for each annotation that is required. See Appendix 1 for the complete JavaScript code, and Appendix 2 for an example of the CRF specific code.

```
acrf_annotate(this, 4, "DM = Demographics", "D", 1);
```

This call creates a domain level annotation on page 4 of the CRF, using pre-defined colour 1 (see Appendix 1) with the text "DM = Demographics".

```
acrf_annotate(this, 4, "STUDYID", "V", 1);
```

This call creates a variable level annotation on page 4 of the CRF, using pre-defined colour 1 with the text "STUDYID".

To make construction of the above calls to acrf_annotate() easier a simple Excel file can be used. (see Figure 3)

	A	B	C	D	E	F
1	Page	Type	Label	Colour	Notes	function acrf_annotate_CDISC()
2	4	D	DM = Demographics		1	acrf_annotate(this, 4, "DM = Demographics", "D", 1);
3	4	V	STUDYID		1	acrf_annotate(this, 4, "STUDYID", "V", 1);
4	4	D	SC = Subject Characteristics		2	acrf_annotate(this, 4, "SC = Subject Characteristics", "D", 2);
5	4	V	SCDTC		2	acrf_annotate(this, 4, "SCDTC", "V", 2);
6						}
7						app.addMenuItem({cName:"Annotate - CDISC", cParent:"Annotated CRF", cExec:"acrf_annotate_CDISC()", nPos:1});
8						

Figure 3: Example Excel sheet used to construct JavaScript code for annotations

Column F is constructed using the Excel CONCATENATE function, and the resulting text copied to a JavaScript file. (See Appendix 2 for example)

Using this approach the annotations shown in Figure 1 have been re-created below. (see Figure 4)

DM = Demographics
SC = Subject Characteristics

CDISC STUDYID

Study: CDISC01

Screening

Assessment Date SCDTC

___/___/___

DEMOGRAPHY

Date of Birth: ___/___/___ BRTHDTC

Gender: Male Female SEX

Ethnicity: Hispanic or Latino Not Hispanic or Latino ETHNIC

Race: Check all that apply RACE

RACE, when more than one selected, RACE-MULTIPLE and individual responses are RACE1, RACE2, etc. in SUPPDM

White

American Indian or Alaska Native

Black or African American

Native Hawaiian or Other Pacific Islander

Asian

Other: RACBOTH in SUPPDM

FamilyStatus: Never Married Domestic Partner CORRES where SCTRESTCD = MARISTAT

Married Divorced

Legally Separated Widowed

Figure 4: Annotations created using the JavaScript code

PhUSE 2015

CREATING BOOKMARKS

The SDTM-MSG suggests creating dual-bookmarks. However, this in effect doubles the required work, and can lead to inconsistencies between the 2 bookmark types if care is not taken. By creating the bookmarks using JavaScript this can be minimized. The code presented only requires a single function to create both bookmarks.

```
_root = obj.bookmarkRoot;

// create/find Visits root
_idx = acrf_find_bookmark(_root, "Visits");
if (_idx == null) {
    acrf_create_bookmark(_root, "Visits", 1, 0);
    _idx = acrf_find_bookmark(_root, "Visits");
}
_visits = obj.bookmarkRoot.children[_idx];

// create/find bookmark for _visit
_idx = acrf_find_bookmark(_visits, _visit);
if (_idx == null) {
    if (_visits.children == null) {
        acrf_create_bookmark(_visits, _visit, _page, 0);
    } else {
        acrf_create_bookmark(_visits, _visit, _page, _visits.children.length);
    }
    _idx = acrf_find_bookmark(_visits, _visit);
}
_bmk = _visits.children[_idx];
if (_bmk.children == null) {
    acrf_create_bookmark(_bmk, _module, _page, 0);
} else {
    acrf_create_bookmark(_bmk, _module, _page, _bmk.children.length);
}
```

Adobe Acrobat represents bookmarks in a tree structure, both in the display when viewing the document and in its internal document model. In the above code, the root bookmark is first obtained. This always exists in the PDF document, although if there are no bookmarks present it will have no child bookmarks. The required three levels of bookmarks are then created:

- Find root bookmark
- Find/Create "Visits" bookmark (level 1)
- Find/Create visit bookmark (level 2)
- Create the CRF form (domain) bookmark (level 3)

This gives the three levels of bookmarks that can be seen in Figure 2.

Creating the second bookmark, with top-level bookmark Domains uses similar code, with the additional requirement that the level 2 bookmarks be sorted alphabetically:

- Find root bookmark
- Find/Create "Domains" bookmark (level 1)
- Find/Create CRF form (domain) bookmark, sorted alphabetically (level 2)
- Create the Visit bookmark (level 3)

As with the annotation creation a simple Excel file can be used to specify the bookmarks and create the JavaScript code.

PhUSE 2015

	A	B	C	D	E
1	Visit	Module	Page	Visit	function acrf_bookmark_CDISC()
2	Screening	Enrollment	1	Screening	acrf_bookmark(this,"Screening","Enrollment", 1);
3		Inclusion Criteria	2	Screening	acrf_bookmark(this,"Screening","Inclusion Criteria", 2);
4		Exclusion Criteria	3	Screening	acrf_bookmark(this,"Screening","Exclusion Criteria", 3);
5	Baseline	Randomization	14	Baseline	acrf_bookmark(this,"Baseline","Randomization", 14);
6		Vital Signs	9	Baseline	acrf_bookmark(this,"Baseline","Vital Signs", 9);
7		Mini-Mental State Examination	11	Baseline	acrf_bookmark(this,"Baseline","Mini-Mental State Examination", 11);
8		Cornell Scale for Depression in Dementia	12	Baseline	acrf_bookmark(this,"Baseline","Cornell Scale for Depression in Dementia", 12);
9	Week 2	Vital Signs	9	Week 2	acrf_bookmark(this,"Week 2","Vital Signs", 9);
10		Laboratory	9	Week 2	acrf_bookmark(this,"Week 2","Laboratory", 9);
11		Cornell Scale for Depression in Dementia	12	Week 2	acrf_bookmark(this,"Week 2","Cornell Scale for Depression in Dementia", 12);
12		Clinical Global Impression	15	Week 2	acrf_bookmark(this,"Week 2","Clinical Global Impression", 15);
13	Week 24	Physical Examination	8	Week 24	acrf_bookmark(this,"Week 24","Physical Examination", 8);
14		Vital Signs	9	Week 24	acrf_bookmark(this,"Week 24","Vital Signs", 9);
15		Laboratory	9	Week 24	acrf_bookmark(this,"Week 24","Laboratory", 9);
16	End of Study / Termination	Termination	16	End of Study / Termination	acrf_bookmark(this,"End of Study / Termination","Termination", 16);
17	Running Records	Study Medication Inventory	17	Running Records	acrf_bookmark(this,"Running Records","Study Medication Inventory", 17);
18		Medication Record	18	Running Records	acrf_bookmark(this,"Running Records","Medication Record", 18);
19		Adverse Events	19	Running Records	acrf_bookmark(this,"Running Records","Adverse Events", 19);
20		Prior/Concomitant Medications	20	Running Records	acrf_bookmark(this,"Running Records","Prior/Concomitant Medications", 20);
21					}
22					app.addMenuItem({Name:"Bookmark CDISC", cParent:"Annotated CRF", cExec:"acrf_bookmark_CDISC()", nPos:1});
23					

Figure 5: Example Excel sheet used to construct JavaScript code for bookmarks

Columns A, B and C are populated as required for the CRF, with columns D and E derived with Excel functions. The resulting JavaScript code in column E is copied to a JavaScript file. (See Appendix 2 for example)

CONCLUSION

Creating the blankcrf.pdf can be a laborious, manual process that can take a long time, and be prone to errors. By extending Adobe Acrobat with JavaScript code this process can be greatly simplified. This allows for easier creation of the blankcrf.pdf in a consistent way, and by implementing the optional suggestions of colour coding domain annotations greatly increases the ease of use for the reviewer.

REFERENCES

Hufford, Walter. 2014 "Automating Production of the blankcrf.pdf". PhUSE 2014, Paper #CC01

"JavaScript™ for Acrobat® API Reference",
http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/js_api_reference.pdf

"Study Data Tabulation Model Metadata Submission Guidelines (SDTM-MSG)", <http://www.cdisc.org/metadata-submission-guideline-%28msg%29-package-preface>

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,
 Oxfordshire,
 OX14 4RY,
 UK
 jn@s-cubed.co.uk
 http://www.s-cubed.co.uk

Brand and product names are trademarks of their respective companies.

PhUSE 2015

APPENDIX 1: JAVASCRIPT CODE – COMMON CODE

The following is the main JavaScript code and should be saved as a single file in the Adobe Acrobat JavaScripts folder.

```
// Main Annotation Procedure
function acrf_annotate(obj, _page, _text, _type, _colour) {
    var annot; // the annotation
    var rt;    // holder for the text with formatting
    var fsz;   // font size
    var fgc;   // text colour
    var bgc;   // fill colour
    var x1;    // left - annotation bounding box
    var x2;    // right - annotation bounding box
    var y1;    // bottom - annotation bounding box
    var y2;    // top - annotation bounding box

    // depending on annotation type - dataset/variable - different font size/colour
    if (_type == "D") {
        fsz = 12;
        fgc = color.black;
    } else {
        fsz = 8;
        fgc = color.red;
    }

    // determine position (hard-coded) and size of bounding box (dependant on text length)
    x1 = 100;
    y1 = 400;
    x2 = x1 + (_text.length+1)*fsz*0.6;
    y2 = y1 + fsz + 2;

    // lookup RGB colour definition
    bgc = acrf_color(_colour);

    // construct the array for the text with formatting
    rt = new Array();
    rt[0] = new Object();
    rt[0].text = _text;
    rt[0].textColor = fgc;
    rt[0].textSize = fsz;
    rt[0].fontFamily = ["Courier", "monospace"];

    // create the annotation
    annot = obj.addAnnot({
        page: _page-1,
        type: "FreeText",
        rect: [x1, y1, x2, y2],
        width: 1,
        strokeColor: color.black,
        fillColor: bgc,
        alignment: 1
    });
    annot.richContents = rt;
    annot.author = "Your Company Name";
}

// Remove Annotations
function acrf_remove_annotations() {
    var annots;

    annots = this.getAnnots();
    if (annots != null) {
        for (var i = 0; i < annots.length; i++) {
            annots[i].destroy();
        }
    }
}

// Main Bookmark Creation Procedure
function acrf_bookmark(obj, _visit, _module, _page) {
    var _root;
    var _visits;
    var _domains;
```

PhUSE 2015

```
var _idx;
var _bmk;

_root = obj.bookmarkRoot;

// create/find Visits root
_idx = acrf_find_bookmark(_root, "Visits");
if (_idx == null) {
    acrf_create_bookmark(_root, "Visits", 1, 0);
    _idx = acrf_find_bookmark(_root, "Visits");
}
_visits = obj.bookmarkRoot.children[_idx];

// create/find bookmark for _visit
_idx = acrf_find_bookmark(_visits, _visit);
if (_idx == null) {
    if (_visits.children == null) {
        acrf_create_bookmark(_visits, _visit, _page, 0);
    } else {
        acrf_create_bookmark(_visits, _visit, _page, _visits.children.length);
    }
    _idx = acrf_find_bookmark(_visits, _visit);
}
_bmk = _visits.children[_idx];
if (_bmk.children == null) {
    acrf_create_bookmark(_bmk, _module, _page, 0);
} else {
    acrf_create_bookmark(_bmk, _module, _page, _bmk.children.length);
}
_bmk.open = false;

// create/find Domains root
_idx = acrf_find_bookmark(_root, "Domains");
if (_idx == null) {
    acrf_create_bookmark(_root, "Domains", 1, 1);
    _idx = acrf_find_bookmark(_root, "Domains");
}
_domains = obj.bookmarkRoot.children[_idx];

// create/find bookmark for _domain
_idx = acrf_find_bookmark(_domains, _module);
if (_idx == null) {
    if (_domains.children == null) {
        // no domains yet - create as position 0
        acrf_create_bookmark(_domains, _module, _page, 0);
    } else {
        // domains exist - iterate to find correct position
        _idx = _domains.children.length;
        for (var i = 0; i < _domains.children.length; i++) {
            if (_domains.children[i].name > _module) {
                _idx = i;
                break;
            }
        }
        acrf_create_bookmark(_domains, _module, _page, _idx);
    }
    _idx = acrf_find_bookmark(_domains, _module);
}
_bmk = _domains.children[_idx];
if (_bmk.children == null) {
    acrf_create_bookmark(_bmk, _visit, _page, 0);
} else {
    acrf_create_bookmark(_bmk, _visit, _page, _bmk.children.length);
}
_bmk.open = false;
}

// create a bookmark
function acrf_create_bookmark(_parent, _text, _page, _location) {
    _parent.createChild(_text, "this.pageNum=" + (_page - 1), _location);
}

// find a bookmark by TEXT label
```

PhUSE 2015

```
function acrf_find_bookmark(_parent, _text) {
  if (_parent.children == null) {
    return null;
  }
  else {
    for (var i = 0; i < _parent.children.length; i++) {
      if (_parent.children[i].name == _text) {
        // bookmark found - return position
        return i;
      }
    }
    // bookmark not found
    return null;
  }
}

// Remove Bookmarks
function acrf_remove_bookmarks() {
  this.bookmarkRoot.remove();
}

// Define array of background colours that work well with black or red text
function acrf_color(_colour) {
  var col;
  col = new Array();
  col[0] = "RGB";

  switch (_colour) {
    case 1: col[1] = 1;    col[2] = 1;    col[3] = 0.75; break;
    case 2: col[1] = 0.75; col[2] = 1;    col[3] = 1;    break;
    case 3: col[1] = 0.75; col[2] = 1;    col[3] = 0.6;  break;
    case 4: col[1] = 1;    col[2] = 0.75; col[3] = 0.75; break;
    case 5: col[1] = 0.75; col[2] = 0.75; col[3] = 1;    break;
    case 6: col[1] = 0.75; col[2] = 0.75; col[3] = 0.4;  break;
    default: col[1] = 1;    col[2] = 1;    col[3] = 1;
  }

  return col;
}

// Add new entries to the File menu
app.addSubMenu({cName:"Annotated CRF", cParent:"File", nPos:1});
app.addMenuItem({cName:"Remove Annotations",cParent:"Annotated
CRF",cExec:"acrf_remove_annotations();" ,nPos:1});
app.addMenuItem({cName:"Remove Bookmarks",cParent:"Annotated
CRF",cExec:"acrf_remove_bookmarks();" ,nPos:1});
```

APPENDIX 2: JAVASCRIPT CODE – CDISC EXAMPLE

The following is the generated JavaScript code to duplicate the example blankcrf.pdf annotations and bookmarks from the SDTM-MSG. It should be saved in the Adobe Acrobat JavaScripts folder in its own file. Note, that as the CRF specific code is split into a separate file to the common code that care should be taken in the names of the two JavaScript files so that Adobe Acrobat can process them in the correct order.

```
function acrf_annotate_CDISC(){
acrf_annotate(this, 4, "DM = Demographics", "D", 1);
acrf_annotate(this, 4, "STUDYID", "V", 1);
acrf_annotate(this, 4, "BRTHDTC", "V", 1);
acrf_annotate(this, 4, "SEX", "V", 1);
acrf_annotate(this, 4, "ETHNIC", "V", 1);
acrf_annotate(this, 4, "RACE", "V", 1);
acrf_annotate(this, 4, "RACEOTH in SUPPDM", "V", 1);
acrf_annotate(this, 4, "RACE, when more than one selected, RACE=MULTIPLE and individual responses
are RACE1, RACE2, etc. in SUPPDM", "V", 1);
acrf_annotate(this, 4, "SC = Subject Characteristics", "D", 2);
acrf_annotate(this, 4, "SCDTC", "V", 2);
acrf_annotate(this, 4, "SCORRES where SCTESTCD = MARISTAT", "V", 2);
}
app.addMenuItem({cName:"Annotate - CDISC", cParent:"Annotated CRF",
cExec:"acrf_annotate_CDISC();" , nPos:1});

function acrf_bookmark_CDISC(){
acrf_bookmark(this,"Screening","Enrollment", 1);
```

PhUSE 2015

```
acrf_bookmark(this,"Screening","Inclusion Criteria", 2);
acrf_bookmark(this,"Screening","Exclusion Criteria", 3);
acrf_bookmark(this,"Screening","Demography", 4);
acrf_bookmark(this,"Screening","Informed Consent", 4);
acrf_bookmark(this,"Screening","Psychiatric History", 5);
acrf_bookmark(this,"Screening","Medical and Surgical History", 6);
acrf_bookmark(this,"Screening","Psychotropic Drug Treatment History", 7);
acrf_bookmark(this,"Screening","Physical Examination", 8);
acrf_bookmark(this,"Screening","Vital Signs", 9);
acrf_bookmark(this,"Screening","Laboratory", 9);
acrf_bookmark(this,"Screening","12-Lead ECG", 10);
acrf_bookmark(this,"Screening","Mini-Mental State Examination", 11);
acrf_bookmark(this,"Screening","Cornell Scale for Depression in Dementia", 12);
acrf_bookmark(this,"Baseline","Randomization", 14);
acrf_bookmark(this,"Baseline","Vital Signs", 9);
acrf_bookmark(this,"Baseline","Mini-Mental State Examination", 11);
acrf_bookmark(this,"Baseline","Cornell Scale for Depression in Dementia", 12);
acrf_bookmark(this,"Week 2","Vital Signs", 9);
acrf_bookmark(this,"Week 2","Laboratory", 9);
acrf_bookmark(this,"Week 2","Cornell Scale for Depression in Dementia", 12);
acrf_bookmark(this,"Week 2","Clinical Global Impression", 15);
acrf_bookmark(this,"Week 24","Physical Examination", 8);
acrf_bookmark(this,"Week 24","Vital Signs", 9);
acrf_bookmark(this,"Week 24","Laboratory", 9);
acrf_bookmark(this,"Week 24","12-Lead ECG", 10);
acrf_bookmark(this,"Week 24","Mini-Mental State Examination", 11);
acrf_bookmark(this,"Week 24","Cornell Scale for Depression in Dementia", 12);
acrf_bookmark(this,"Week 24","Clinical Global Impression", 15);
acrf_bookmark(this,"End of Study / Termination","Termination", 16);
acrf_bookmark(this,"Running Records","Study Medication Inventory", 17);
acrf_bookmark(this,"Running Records","Medication Record", 18);
acrf_bookmark(this,"Running Records","Adverse Events", 19);
acrf_bookmark(this,"Running Records","Prior/Concomitant Medications", 20);
}
app.addMenuItem({cName:"Bookmark CDISC", cParent:"Annotated CRF", cExec:"acrf_bookmark_CDISC()",
nPos:1});
```