

Extracting Useful Information from XFDF Format of aCRF Annotations

Tingting Zhang, Yuan Sun, Sanofi

ABSTRACT

The annotated Case Report Form (aCRF) describes the collected data in context by annotating the corresponding SDTM datasets, variables, and any associated notes identified on the CRF. It is important to ensure the accuracy of aCRF, but it is time-consuming to manually cross-check aCRF and DEFINE.XML. Adjusting the width of annotations for NMPA submission after translating the English CRF into Chinese version has the same issue.

This paper introduces how to extract the information from XFDF format of aCRF annotations to streamline this process. Page numbers extracted from XFDF format compare with those in DEFINE.XML help identify inconsistencies. Checking colors ensures the uniformity within each domain. Automatically adjusting annotation widths with position information ensures display the text complete. These operations can easily reduce manual issues and save time.

INTRODUCTION

In clinical trials, the annotated Case Report Form (aCRF) is a crucial document that records all data collection fields and their annotations. The aCRF needs to be consistent with SDTM datasets. If a variable is annotated on the CRF, this variable must be in the SDTM dataset and mapped in the SDTM DEFINE.XML with ORIGIN=CRF. This paper presents a SAS-based approach to efficiently extract annotation information from XFDF files based on the blankcrf.pdf file in Metadata Submission Guideline (MSG) for SDTMIG version 1.0¹, because the annotations in acrf.pdf in MSG version 2.0² are flattened, can't export the XFDF file.

XFDF FORMAT OVERVIEW

XFDF (XML Forms Data Format) is a file format used to store data in XML format, specifically for handling form data in PDF documents. It is an extension of the PDF format used to handle forms and annotations, offering higher readability and ease of handling. To export XFDF file, we can directly click on the annotation, then click on “Export All To Data File...”, and save it as an XFDF format file. Figure 1 provides an example based on blankcrf.pdf in MSG version 1.0.

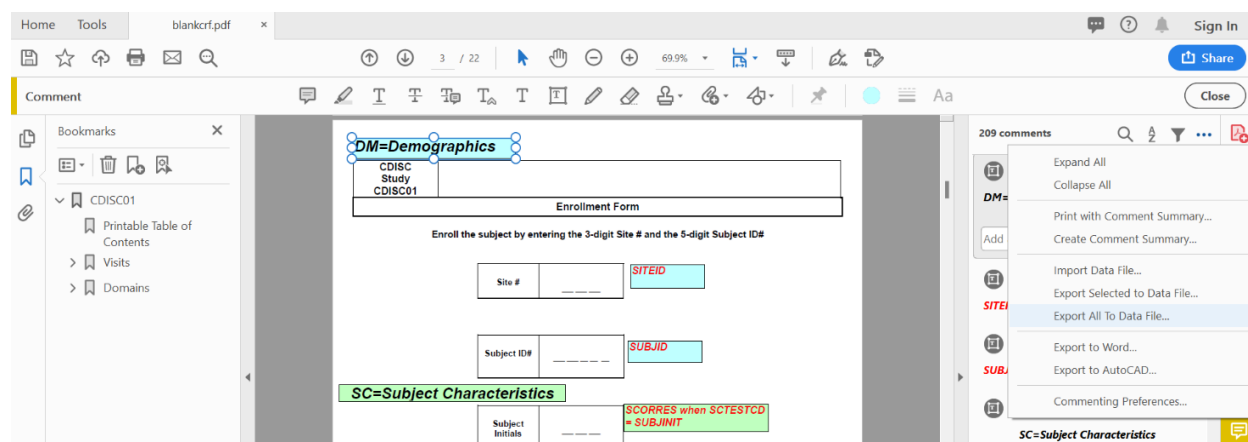


Figure 1. Export XFDF file

Taking “DM=Demographics” annotation as an example, the XFDF format is as follows:

```
<?xml version="1.0"?>
<!-- Root element with XML namespace declaration for XFDF -->
<xfdf xmlns="http://ns.adobe.com/xfdf/" xml:space="preserve">

  <!-- Annotations section -->
  <annots>

    <!-- FreeText annotation -->
    <freetext color="#BFFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20091112084313-05'00'" name="a60dc033-5534-4833-9ae9-8ea63131213a" page="2" fringe="0.500150,0.500150,0.500150,0.500150" rect="21.880200,746.255000,212.123000,771.740000" subject="IE">
      <contents-richtext>
        <body xmlns="http://www.w3.org/1999/xhtml" xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/" xfa:APIVersion="Acrobat:7.0.8" xfa:spec="2.0.2" style="font-size:18.0pt;font-weight:bold;font-style:italic;font-family:Arial">
          <p dir="ltr">DM=Demographics</p>
        </body>
      </contents-richtext>
      <!-- Default appearance settings for the annotation -->
      <defaultappearance>0 0 0 rg /Arial,BoldItalic 18 Tf</defaultappearance>
    </freetext>

    <!-- Default style settings for the annotation -->
    <defaultstyle>font: italic bold Arial 18.0pt; text-align:left; color:#000000</defaultstyle>
  </annots>

  <!-- Reference to the Linked PDF file -->
  <f href="blankcrf.pdf"/>
</xfdf>
```

The “<freetext>” element represents a free text annotation. Its attributes include:

1. color: The color of the annotation (in hexadecimal format)
2. page: The page number where the annotation is located
3. rect: The position and size of the annotation (rectangular coordinates)

And the “<p>” element contains the actual text of the annotation.

EXTRACTING INFORMATION FROM XFDF FILES

As shown above, each annotation is included in one “<freetext>” element, allowing extraction of page, color, position and annotation from the XFDF file.

The extraction process can be broken down into several steps:

1. Import the XFDF file.
2. Identify annotations: annotations are included in annotations section. We should only keep the section and extract free text annotations by identifying the <freetext> elements and concatenate the lines within these elements into a single text string for each annotation.
3. Extract attributes and text content: Parse the annotation content to extract specific attributes and annotation text, this is done use Perl regular expressions functions:

```
pattern_color=prxparse('/color="(.*?)"/');
call prxsubstr(pattern_color, full_content, position_color, length_color);
if position_color ne 0 then color=strip(substr(full_content,position_color+7,length_color-8));
```

Figure 2 shows the information extracted from XFDF file of blankcrf.pdf.

obj	full_content	color	@ page	anno	rect	domain
1	5 ><freetext color="#BFFFFB" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFB	3	SCORRES when SCTESTCD = SUBJINIT	335.467000,431.595000,504.531000,465.405000	SC
2	3 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	3	SUBJID	340.016000,511.744000,427.983000,538.254000	DM
3	2 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	3	SITEID	343.016000,597.243000,430.983000,626.753000	DM
4	11 ><freetext color="#A4FFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#A4FFFF	4	IECAT	159.067000,574.370000,206.341000,590.380000	
5	8 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	4	IEORRES when IETESTCD = INCL01	374.066000,534.078000,586.116000,550.489000	IE
6	12 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	4	IETEST	39.061700,542.724000,107.029000,559.734000	IE
7	7 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	4	IETDC	498.576000,617.746000,544.298000,636.755000	IE
8	6 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	4	VISIT	461.702000,657.056000,503.420000,674.065000	IE
9	10 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	4	IEORRES when IETESTCD = INCL03	371.228000,445.182000,585.027000,467.593000	IE
10	9 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	4	IEORRES when IETESTCD = INCL02	374.888000,499.873000,593.938000,517.284000	IE
11	16 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	5	IEORRES when IETESTCD = EXCL01	374.066000,607.453000,590.116000,625.863000	IE
12	15 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	5	IETDC	494.078000,691.120000,541.298000,711.630000	IE
13	18 ><freetext color="#BFFFFF" creationdate="D:00000000000000Z" flags="print" date="D:20...	#BFFFFF	5	IECAT	155.567000,643.244000,196.841000,660.254000	IE

Figure 2. Information from XFDF File

CHECK COLOR OF ANNOTATIONS

According to MSG, if more than one domain exists on a page, and all its variables should be color-coded. Annotations of domain and its variables should be matched with color and page. The 4th-10th records in Figure 2 show the annotations in Figure 3, the domain for the 4th record is blank, it should be "IE" based on the annotation "IECAT" and page number, but the color code for "IECAT" is different from other variables in IE domain.

Figure 3. aCRF for IE

EXTRACT PAGE FOR DEFINE

The variable names or value level metadata (VLM) for --ORRES variable can be extracted from annotations and used for generating or validating the DEFINE.XML. We can cross-check the variables annotated on CRF with the variables marked as originating from CRF in DEFINE.XML to ensure consistency and verify that the corresponding pages match. The bi-directional consistency between aCRF and DEFINE.XML is important in data submission.

ADJUST ANNOTATION WIDTHS

For NMPA submission, the English CRF should be translated into Chinese, and the annotations for domains and "[NOT SUBMITTED]" should also be translated. Importing an updated XFDF file with translated annotations into the Chinese CRF will save more time than manually updating each annotation. However, there will be a new issue, the annotation rectangles are sized based on the English text, but a Chinese character may wider than an English word. As a result, translated annotations may not display completely in the original rectangles. Therefore, the widths should be adjusted to ensure the complete display of the text.

In XFDF, the "rect" attribute specifies the position and size of an annotation, includes four values that represent the coordinates of the lower-left and upper-right corners of the rectangle. The format is as follows:

rect="x1,y1,x2,y2"

- "x1, y1": Coordinates of the lower-left corner of the rectangle.

- "x2, y2": Coordinates of the upper-right corner of the rectangle.

Figure 4 shows the coordinates of the lower-left corner of "DM=Demographics" annotation, with the page origin at the top-left corner in PDF, the x value increases from right to left, and y value increases from top to bottom, as shown by the red axis in Figure 4. In contrast, the values in "rect" are specified from bottom-left, the y value increases from bottom to top, as shown by the green axis. The height of the page is 792, so the y value in PDF plus the one in "rect" attribute should be 792. Therefore, we need to use the formulas below to calculate the "rect" values:

$x1 = X$ (in black rectangle)

$y1 = 792$ (height of page) – Y (in black rectangle)

$x2 = x1 + W$ (width)

$y2 = y1 + H$ (height)

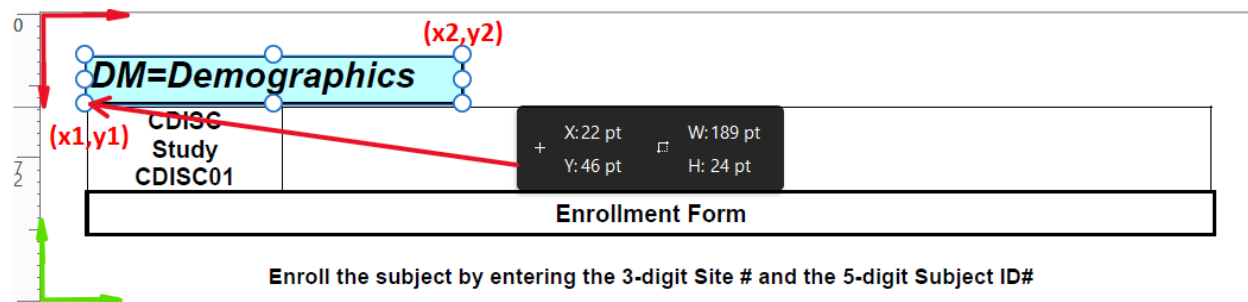


Figure 4. Coordinates of the lower-left corner of "DM=Demographics" annotation in PDF

After calculation, the "rect" value should be "22,746,211,770", ignoring manual errors, it's almost the same as rect="21.880200,746.255000,212.123000,771.740000" from the following codes in XFDF example in section XFDF format overview. And it can match with the coordinates in Figure 5.

```
<freetext color="#BFFFFF" creationdate="D:0000000000000Z" flags="print" date="D:20091112084313-05'00'" name="a60dc033-5534-4833-9ae9-8ea63131213a" page="2" fringe="0.500150,0.500150,0.500150,0.500150" rect="21.880200,746.255000,212.123000,771.740000" subject="IE">
```

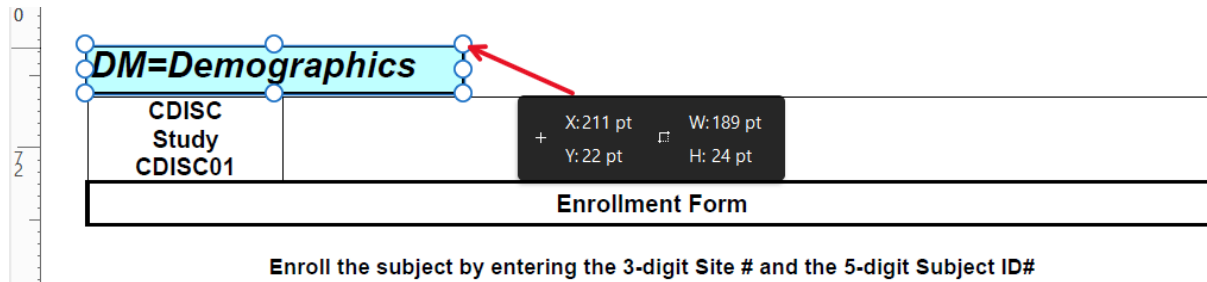


Figure 5. Coordinates of the upper-right corner of "DM=Demographics" annotation in PDF

This example demonstrates how "rect" attribute works, which means we can use the width and height of a single Chinese character to calculate the values for "x2, y2" to update the size of the rectangle to show complete text.

The font for annotations in the example aCRF is Arial, it's a proportional font, the width of English letters varies depending on their shape and design. Typically, Chinese characters are of equal width, with each character occupying a square-shaped box. The units for font size and coordinates are the same. So, we can use 1.3 times the font size as the height of the annotation rectangle, and $0.5 * \text{font size} * (\text{length of annotation} + 1.5)$ as the width. The "1.3" and "1.5" ensure complete display, can be adjusted based on specific requirements.

For example, the font size is 18 points, and the annotation text length is 11 characters for "DM = 人口学", the height of the annotation rectangle would be:

$$H = 1.3 * 18 = 23.4$$

And the width would be:

$$W = 0.5 * 18 * (11 + 1.5) = 112.5$$

Then the updated "x2, y2" value in "rect" attribute would be "134.3802,769.655", as shown in the updated aCRF in Figure 6. The rectangle is appropriately sized to fit the annotation, ensuring it does not extend too far and cover up the text on CRF, nor shrink too much to display the complete annotation.

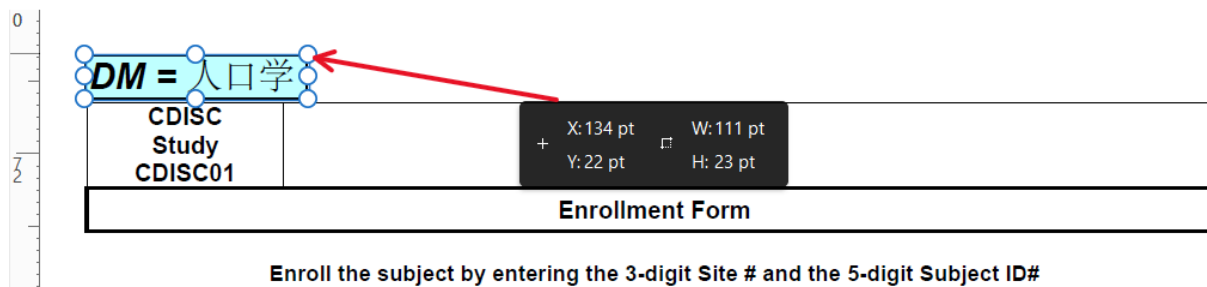


Figure 6. Coordinates of the upper-right corner of "DM = 人口学" annotation in PDF

CONCLUSION

The presented methodology demonstrates an approach to extracting and processing annotation data from XFDF files using SAS. We can significantly enhance the accuracy and efficiency of handling aCRF by extracting the useful information automatically instead of manually.

CODE

EXTRACT INFORMATION FOR DEFINE

```
filename anno "path/blankcrf.xfdf";

data anno;
  infile anno lrecl=32767 length=len;
  input line $varying32767. len;
  linenum+1;
run;

/* Identify annotation section */
data anno1;
  set anno;
  by linenum;
  retain anno 0;
  if index(line,"><annots")>0 then anno=1;
  if index(line,"></annots")>0 then anno+1;
  if anno=1 and line ne "><annots";
  drop anno;
run;

/* Extract freetext and concentrate lines */
data anno2;
  set anno1;
  by linenum;
  retain obj 0;
  if index(line,"><freetext")>0 then obj+1;
run;

proc sort data=anno2;
  by obj linenum;
run;

data anno3;
  set anno2;
  by obj linenum;
  length full_content $2000.;
  retain full_content;

  if first.obj then full_content=line;
  else full_content=catx(' ',full_content,line);

  if last.obj;
run;

/* Extract attributes and annotations */
```

```

data anno4 check_anno;
    set anno3;

    length color page0 anno anno0 anno1 rect $2000. page 8.;
    pattern_color=prxparse('/color="(.*?)"/');
    call prxsubstr(pattern_color, full_content, position_color, length_color);
    if position_color ne 0 then color=strip(substr(full_content,position_color+7,length_color-8));

    pattern_page=prxparse('/page="(.*?)"/');
    call prxsubstr(pattern_page, full_content, position_page, length_page);
    if position_page ne 0 then page0=strip(substr(full_content,position_page+6,length_page-7));
    page=input(page0,best.)+1;

    pattern_anno=prxparse('/<p .*>(.*?)<\/p/');
    call prxsubstr(pattern_anno, full_content, position_anno, length_anno);
    if position_anno ne 0 then anno0=strip(substr(full_content,position_anno+2,length_anno-3));
    anno1=prxchange('s/<[^\>]*>\/', -1, anno0);

    pattern_anno1=prxparse('/>(.*?)<\/');
    call prxsubstr(pattern_anno1, anno1, position_anno1, length_anno1);
    if position_anno1 ne 0 then anno=strip(substr(anno1,position_anno1+1,length_anno1-2));

    pattern_rect=prxparse('/rect="(.*?)"/');
    call prxsubstr(pattern_rect, full_content, position_rect, length_rect);
    if position_rect ne 0 then rect=substr(full_content,position_rect+6,length_rect-7);

    if missing(color) then output check_anno;
    else output anno4;

    drop pattern_: position_: length_;
run;

data anno5;
    set anno4(drop=anno0 anno1);

    length domain $10.;
    if index(anno,"=") then domain=scan(anno,1,"=");
    if length(domain) not in (2,4) then call missing(domain);
run;

proc sql;
    create table anno6 as

```

```

select a.obj, a.full_content, a.color, a.page, a.anno, a.rect, b.domain
from anno5(when=(domain="")) as a
left join anno5(when=(domain ne "")) as b
on a.page=b.page and a.color=b.color
;
quit;

/* Check color */
data check_color;
set anno6;
if domain="";
put "WARNING: Please check color for annotation " anno "on page " page;
run;

/* Extract variable pages */
data var;
set anno6;

length variable0 $100.;

if index(anno,"NOT SUBMITTED")>0 then delete;

if index(anno,"when")>0 then do;
variable0=strip(substr(anno,1,index(anno,"when")-1));
end;
else if index(anno,"=")>0 then do;
variable0=strip(scan(anno,1,"="));
end;
else if index(anno," ")>0 then do;
variable0=strip(scan(anno,1," "));
end;
else variable0=strip(anno);
run;

data var1;
set var;

length variable $200.;
if index(variable0,"/")>0 then do;
do i=1 to count(variable0,"/")+1;
variable=strip(scan(variable0,i,"/"));
output;
end;
end;
else do;
variable=strip(variable0);
output;
end;

```

```

    drop variable0 i;
run;

data var1 check_var;
    set var1;

    if length(variable)<=8 then output var1;
    else output check_var;
run;

proc sort data=var1 out=var2 nodupkey;
    by domain variable page;
run;

data var_page;
    set var2;
    by domain variable page;

    length var_page $2000.;
    retain var_page;
    if first.variable then var_page=strip(page);
    else var_page=strip(var_page||" "||strip(page));

    if last.variable;
run;

/* Extract value level metadata (VLM) for --ORRES */
data orres;
    set var1(where=(index(anno,"ORRES")>0 and index(anno,"TESTCD")>0));

    length testcd0 testcd $200.;
    testcd0=strip(scan(anno,2,"="));

    if index(anno,",")>0 then do;
        do i=1 to count(anno,",")+1;
            testcd=strip(scan(testcd0,i,","));
            output;
        end;
    end;
    else do;
        testcd=strip(testcd0);
        output;
    end;

    drop testcd0 i;
run;

```

```

proc sort data=orres out=orres1 nodupkey;
    by domain variable testcd page;
run;

data orres_page;
    set orres1;
    by domain variable testcd page;

    length orres_page $2000.;
    retain orres_page;
    if first.testcd then orres_page=strip(put(page,best.));
    else orres_page=strip(orres_page)|| " " ||strip(put(page,best.));

    if last.testcd;
run;

```

TRANSLATE ANNOTATION AND ADJUST WIDTHS

```

filename anno "path/blankcrf.xfdf";

/* Translate annotations for domain and "[NOT SUBMITTED]" */
/* Keep only NOT SUBMITTED and AE/CM/DM for example here */
data en2cn;
    infile datalines dsd dlm='|' trunccover ;
    input anno_en :$100. anno_cn :$100.;
    cards;
[NOT SUBMITTED]||[未提交]
AE=Adverse Events|AE = 不良事件
CM=Concomitant Medication|CM = 既往与合并用药
DM=Demographics|DM = 人口学
;
run;

data en2cn1;
    set en2cn;
    anno_en=upcase(anno_en);
    n=_n_;
run;

proc sql noprint;
    select anno_en, anno_cn into: anno_en1- , :anno_cn1- from en2cn1;
    select max(n) into: n trimmed from en2cn1;
run;

%put &&anno_en&n.;

data acrf;

```

```

infile anno lrecl=32767 length=len termstr=LF;
input line $varying32767. len;
linenum+1;
run;

data acrf1;
  set acrf;
  by linenum;
  retain obj 0;
  if index(line,"><freetext")>0 then obj+1;
run;

%macro en2cn();
data acrf2;
  set acrf1;
  length line1 $32767.;
  line1=uppercase(tranwrd(line," = ","=")); ** To avoid different wording, e.g. uppercase or lowercase, spaces **;
  if index(line1,"IF ")=0 and index(line1,"WHEN")=0 and index(line1,"WHERE")=0 and index(line1,"RECT=")=0 then do;
    %do i=1 %to &n.;
      if (index(line1,"&&anno_en&i")>0 or index(line1,"&&anno_cn&i")>0) then do;
        line=tranwrd(line1,"&&anno_en&i","&&anno_cn&i");

        if index(line,"</P") then line=tranwrd(line,"</P","</p");
        if index(line,"</SPAN") then line=tranwrd(line,"</SPAN","</span");

        flg="Y";
        len=length("&&anno_cn&i");
      end;
    %end;
  end;
run;

proc sort data=acrf2;
  by obj linenum;
run;

%mend;

%en2cn;

data rect anno;
  set acrf2;
  by obj linenum;
  if first.obj then output rect;
  else output anno;

```

```

run;

/* Extract font size */
data fontsz;
    set acrf2;

    length fontsz 8.;
    pattern_fontsz=prxparse('/font-size:(.*?);/');
    call prxsubstr(pattern_fontsz, line, position_fontsz, length_fontsz);
    if position_fontsz ne 0 then fontsz=input(substr(line,position_fontsz+10,length_fontsz-13),best.);

    if fontsz ne .;

    keep obj fontsz;
run;

/* Adjust annotation widths */
data rect1;
    merge rect(in=a) anno(where=(flg='Y') keep=obj flg len) fontsz;
    by obj;

    /* Get the location of annotation rectangle */
    length rect $2000.;
    pattern_rect=prxparse('/rect="(.*?)"');
    call prxsubstr(pattern_rect, line, position_rect, length_rect);
    if position_rect ne 0 then rect=substr(line,position_rect+6,length_rect-7);

    /* Lower Left corner */
    llow_x=input(scan(rect,1,","),best.);
    llow_y=input(scan(rect,2,","),best.);

    /* Extract rotation degree to take care annotation with rotation */
    pattern_rot=prxparse('/rotation="(.*?)"');
    call prxsubstr(pattern_rot, line, position_rot, length_rot);
    if position_rot ne 0 then rot=strip(substr(line,position_rot+10,length_rot-11));

    /* Calculate upper right corner */
    if flg='Y' then do;
        /* 0.5 * font size * (length of annotation + 1.5) as width (without rotation) */
        /* 1.3 * font size as height (without rotation) */
        /* Exchange the position of rupp_x and rupp_y due to rotation */
        if rot="90" then rect1=catx(" ",llow_x,llow_y,llow_x+(fontsz*1.3),llow_y+(fontsz*0.5*(len+1.5)));
        else rect1=catx(" ",llow_x,llow_y,llow_x+(fontsz*0.5*(len+1.5)),llow_y+(fontsz*1.3));
    end;
run;

```

```

        line=tranwrd(line,trim(rect),trim(rect1));
    end;
run;

data acrf3;
    set anno rect1;
    keep line linenum obj;
run;

proc sort data=acrf3;
    by linenum;
run;

```

REFERENCES

1. CDISC SDS Metadata Team. "Metadata Submission Guideline (MSG) for SDTMIG". 2011. Available at <https://www.cdisc.org/standards/foundational/sdtmig/metadata-submission-guideline-msg-sdtmig>
2. CDISC SDS MSG Team. "SDTM Metadata Submission Guidelines v2.0". 2021. Available at <https://www.cdisc.org/standards/foundational/sdtm/sdtm-metadata-submission-guidelines-v2-0>

CONTACT INFORMATION

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

Tingting Zhang
 Sanofi
Tingting5.Zhang@Sanofi.com