PharmaSUG China 2021 - Paper AD-035

# CRF Chinese Translation – Mask Plan

Chunpeng Zhao, Liming Xie, Chunling Xu, BeiGene

## ABSTRACT

According to the National Medical Products Administration (NMPA) Guideline on the submission of clinical trial data effective from October 1st, 2020, we need to translate foreign language CRFs into Chinese. This paper will introduce the CRF translation solution that named as Mask Plan.

Through SAS® Procedure Groovy, we run Java code in PrintTextLocations.java from Apache® PDFBox to extract each letter and its coordinate in its page in CRF. With SAS®, we concatenate letters into words, then into sentences. We extract each English string and its location coordinates in its CRF page. We call this English string and its located area as English Face.

We create an annotation box for each English string. The annotation box has opaque white background color. Its size and coordinates in the page are as exactly same as the English string. Its content is the English string's Chinese translation. Then we create a Chinese Mask for the English string. We only see Chinese translation covering on the English string. Underneath the Chinese Mask, English Face is invisible. When we import SDTM annotations from SDTM aCRF English version into Chinese version, we do not need adjust SDTM annotations' sizes or locations, since each English string is translated in place.

## INTRODUCTION

Most of CRF fields (about 80%, even more) in a clinical trial are copied directly from other trials. English strings in a CRF and their Chinese translations are the same with reference trials. The paper will introduce an approach to recycle each English string's Chinese translation, making a road to cheaper CRF English translation.

The sole purpose of life has been to pass on what was learned (Professor Norman, 2014). Let us see how to pass on what was learned on each English string's Chinese translation from one trial CRF to another, automatically and dynamically, just like passing on DNA in organism from one generation to next.

## EXTRACT ENGLISH STRINGS AND THEIR COORDINATES IN CRF PAGES

Through Apache® PDFBox Java code in SAS® Procedure Groovy and SAS code, read each CRF English string and its coordinates in its page. See Yin-Jhen Yan etc., 2018 for details. Here are instructions how to read each letter and its coordinates from CRF based on our practice. You could follow it blindly without worrying about SAS® Procedure Groovy or Apache® PDFBox Java code anymore if you are not familiar with them at all.

1. Download pdfbox-app-2.0.24.jar and pdfbox-2.0.24-src.zip from https://pdfbox.apache.org/download.html.
2. Unzip pdfbox-2.0.24-src.zip to Folder pdfbox-2.0.24-src.
3. Update below Macro `m_PrintTextLocations.`
    a. Two Yellow highlighted `YourWorkingFolderPath` should be updated into your working folder path, which pdfbox-app-2.0.24.jar and Folder pdfbox-2.0.24-src are located.
    b. Assign Macro argument PDF with your CRF folder path and file name.
    c. Assign Macro argument OutDataSet with your defined output SAS dataset name.

```
%macro m_PrintTextLocations(PDF=, OutDataSet=);
```

```
filename javaloc "YourWorkingFolderPath/pdfbox-2.0.24-src/pdfbox-
2.0.24/examples/src/main/java/org/apache/pdfbox/examples/util/PrintText
Locations.java";

data javaloc;
  length STRING $2000;
  retain OutFlag InsertFlag InsertLocation;
  infile javaloc length=l lrecl=2000 end=eof;
  input STRING $varying2000.l;

  if _N_=1 then
    do;
      STRING='proc groovy CLASSPATH="'||'YourWorkingFolderPath/pdfbox-
app-2.0.24.jar'||'";';
      output;
      STRING='  submit;';
      output;
    end;

  if cats(String)="import java.io.ByteArrayOutputStream;" then
    OutFlag=1;
  if OutFlag=1 then
    output;

  InsertLocation+1;
  if cats(String)="public static void main( String[] args ) throws
IOException" then
    do;
      InsertFlag=1;
      InsertLocation=0;
    end;
  if InsertFlag=1 and InsertLocation=1 then
    do;
      String="      args = new String[1];";
      output;
      String='      args[0] = "'||"&PDF"||'";';
      output;
    end;


  if eof then
    do;
      String="  endsubmit;";
      output;
      String="quit;";
      output;
    end;

run;

data _null_;
  set javaloc;
  FILE "%sysfunc(pathname(work))/javaloc.sas" old;
  l=length(String);
  put STRING $varying2000.l;
run;
```

```
filename rawcrf "%sysfunc(pathname(work))/rawcrf.txt";
proc printto log=rawcrf new;
run;

%include "%sysfunc(pathname(work))/javaloc.sas";

proc printto;
run;

data &OutDataSet;
  length STRING $2000 CHAR $1;
  infile rawcrf length=l lrecl=2000 end=eof;
  input STRING $varying2000.l;
  if prxmatch("/^String/", STRING);
  Char=substr(String, find(String, "]")+1);
  X=input(scan(STRING, 2, ",[] "), best.);
  Y=input(scan(STRING, 3, ",[] "), best.);
  FS=input(scan(scan(STRING, 4, ",[] "), 2, "="), best.);
  XSCALE=input(scan(scan(STRING, 5, ",[] "), 2, "="), best.);
  HEIGHT=input(scan(scan(STRING, 6, ",[] "), 2, "="), best.);
  SPACE=input(scan(scan(STRING, 7, ",[] "), 2, "="), best.);
  WIDTH=input(scan(scan(STRING, 8, ",[] "), 2, "="), best.);
run;
%mend m_PrintTextLocations;

%m_PrintTextLocations(PDF=YourCRFFilePathandFileName.pdf,
OutDataSet=YourDefinedOutputDatasetName)
```

4. Run `%m_PrintTextLocations` with your updated working folder path and specified values of Macro arguments PDF and OutDataSet. All letters (CHAR) in a PDF file with their coordinates (X, Y) in its page and their attributes (e.g., Font Size (FS), Height and Width) from left to right, from top to bottom, and from page to next will be extracted into SAS dataset as below. X/Y are coordinates of left top conner of a letter. See Using PDFBox to locate text coordinates within a PDF in Java for details. Each CRF page's origin of coordinate (0, 0) is at the left top conner.

| CHAR | X | Y | FS | HEIGHT | WIDTH |
|---|---|---|---|---|---|
| S | 246.249 | 97.893005 | 9 | 6.6150002 | 5.7059937 |
| u | 251.95499 | 97.893005 | 9 | 6.6150002 | 5.5620117 |
| b | 257.517 | 97.893005 | 9 | 6.6150002 | 5.471985 |
| j | 262.98898 | 97.893005 | 9 | 6.6150002 | 2.9519958 |
| e | 265.94098 | 97.893005 | 9 | 6.6150002 | 5.192993 |
| c | 271.13397 | 97.893005 | 9 | 6.6150002 | 4.5539856 |
| t | 275.68796 | 97.893005 | 9 | 6.6150002 | 3.402008 |
| | 279.08997 | 97.893005 | 9 | 6.6150002 | 3.0509949 |
| C | 282.14096 | 97.893005 | 9 | 6.6150002 | 6.048004 |
| a | 288.18896 | 97.893005 | 9 | 6.6150002 | 5.220001 |
| s | 293.40897 | 97.893005 | 9 | 6.6150002 | 4.4819946 |
| e | 297.89096 | 97.893005 | 9 | 6.6150002 | 5.192993 |
| | 303.08395 | 97.893005 | 9 | 6.6150002 | 3.0509949 |
| R | 306.13495 | 97.893005 | 9 | 6.6150002 | 6.1380005 |
| e | 312.27295 | 97.893005 | 9 | 6.6150002 | 5.192993 |

| | | | | | |
|---|---|---|---|---|---|
| p | 317.46594 | 97.893005 | 9 | 6.6150002 | 5.471985 |
| o | 322.93793 | 97.893005 | 9 | 6.6150002 | 5.3370056 |
| r | 328.27493 | 97.893005 | 9 | 6.6150002 | 3.7170105 |
| t | 331.99194 | 97.893005 | 9 | 6.6150002 | 3.402008 |

Based on your CRF layout settings (e.g., paragraph spacing, line spacing, word spacing…), concatenate letters into words, then into sentences with your SAS code. For example, we can read each string and its coordinates from CRF in Figure 1 into SAS dataset in Table 1.

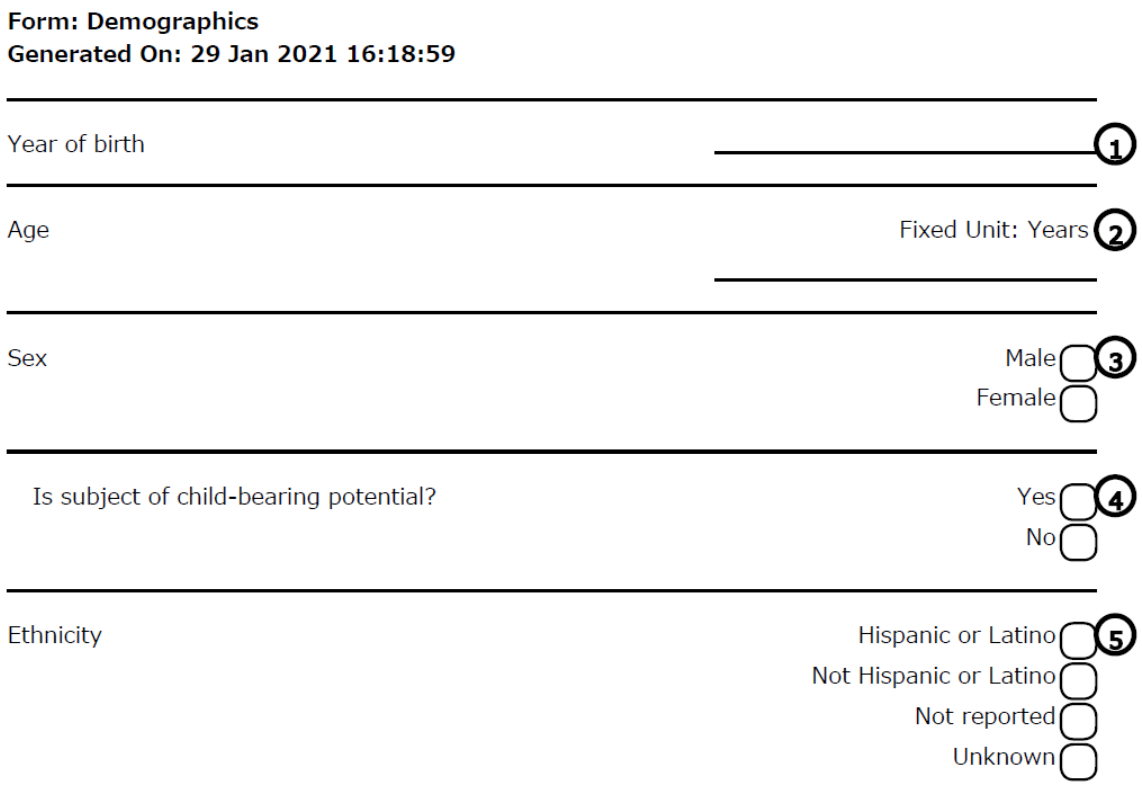


**Figure 1. Sample of CRF Fields**

**Table 1 Sample of CRF Strings and Their Coordinates**

| Page | CRFForm | Field SEQ | FieldLabel | FieldOption | FieldLabel STX | Fieldlabel ENX | FieldOption STX | FieldOption ENX |
|---|---|---|---|---|---|---|---|---|
| 11 | Demographics | | Form: Demographics Generated On: 29 Jan 2021 16:18:59 | | 90 | 268.5420533 | | |
| 11 | Demographics | 1 | Year of birth | | 90 | 144.6839751 | | |
| 11 | Demographics | 2 | Age | Fixed Unit: Years | 90 | 106.7400057 | 443.877 | 518.9999851 |
| 11 | Demographics | 3 | Sex | Male | 90 | 105.894005 | 485.84 | 506.000003 |
| 11 | Demographics | 3 | Sex | Female | 90 | 105.894005 | 474.347 | 506.0000146 |
| 11 | Demographics | 4 | Is subject of child-bearing potential? | Yes | 100 | 260.1819146 | 490.655 | 506.000001 |
| 11 | Demographics | 4 | Is subject of child-bearing potential? | No | 100 | 260.1819146 | 493.994 | 506.0000119 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 11 | Demographics | 5 | Ethnicity | Hispanic or Latino | 90 | 127.673996 | 427.511 | 506.0000438 |
| 11 | Demographics | 5 | Ethnicity | Not Hispanic or Latino | 90 | 127.673996 | 409.052 | 506.0000586 |
| 11 | Demographics | 5 | Ethnicity | Not reported | 90 | 127.673996 | 450.038 | 506.0000054 |
| 11 | Demographics | 5 | Ethnicity | Unknown | 90 | 127.673996 | 464.96 | 506.0000696 |

| Page | CRFForm | Field SEQ | FieldLabel | FieldOption | Y | YEnd | OptionYST | OptionY |
|---|---|---|---|---|---|---|---|---|
| 11 | Demographics | 0 | Demographics | | 97.893005 | 136.893 | | |
| 11 | Demographics | 1 | Year of birth | | 172.893 | 174.893 | 172.893 | 174.893 |
| 11 | Demographics | 2 | Age | Fixed Unit: Years | 206.893 | 208.893 | 206.893 | 208.893 |
| 11 | Demographics | 3 | Sex | Male | 257.393 | 273.393 | 257.393 | 259.393 |
| 11 | Demographics | 3 | Sex | Female | 257.393 | 273.393 | 273.393 | 273.393 |
| 11 | Demographics | 4 | Is subject of child-bearing potential? | Yes | 312.393 | 328.393 | 312.393 | 314.393 |
| 11 | Demographics | 4 | Is subject of child-bearing potential? | No | 312.393 | 328.393 | 328.393 | 328.393 |
| 11 | Demographics | 5 | Ethnicity | Hispanic or Latino | 367.393 | 415.393 | 367.393 | 369.393 |
| 11 | Demographics | 5 | Ethnicity | Not Hispanic or Latino | 367.393 | 415.393 | 383.393 | 383.393 |
| 11 | Demographics | 5 | Ethnicity | Not reported | 367.393 | 415.393 | 399.393 | 399.393 |
| 11 | Demographics | 5 | Ethnicity | Unknown | 367.393 | 415.393 | 415.393 | 415.393 |

For example, in Table 1, String Sex rectangular box coordinates are (90, 257, 106, 259), i.e., (Field label start X (FieldLabelSTX), Field label start Y (OptionYST), Field label end X (FieldlabelENX), Field label end Y (OptionY)). In practice, field label and its first option share the same Y coordinates. Each option's rectangular box coordinates are (Field option start X (FieldOptionSTX), Field option start Y (OptionYST), Field option end X (FieldOptionENX), Field option end Y (OptionY)). And the whole field's rectangular box coordinates are (Field label start X (FieldLabelSTX), Field label start Y (Y), Field option end X (FieldOptionENX), the last Field option end Y (YEnd)).

## CENTRALIZED TRANSLATION DICTIONARY

We create a centralized translation dictionary. Each CRF file has its translation dataset as below. All translation datasets share the same data structure. All CRF files can reference each other on translation. One English string only needs to be translated once. New CRF will focus on new English strings' translation. For example, in Table 2, English Strings appearing at Page 6 at first time in CRF File BGB1234_101_V3, their Chinese translations reference CRF Global_Common_V2_1, except String "Is subject of child-bearing potential?". It means in CRF File BGB1234_101_V3 Page 6, only "Is subject of child-bearing potential?" needs to be translated. All others were translated in CRF File Global_Common_V2_1 already. Their Chinese translations can be copied into CRF File BGB1234_101_V3 translation dictionary automatically.

**Table 2 Sample of Translation Dictionary**

| CRFID | StringEN | StringCN | Firstpage | RefCRFID | RefFirstPage |
|---|---|---|---|---|---|
| BGB1234_101_V3 | Form: Demographics | 表格：人口统计 | 6 | Global_Common_V2_1 | 6 |
| BGB1234_101_V3 | Year of birth | 出生年份 | 6 | Global_Common_V2_1 | 6 |

| | | | | | |
|---|---|---|---|---|---|
| BGB1234_101_V3 | Age | 年龄 | 6 | Global_Common_V2_1 | 6 |
| BGB1234_101_V3 | Fixed Unit: Years | 固定单位：年 | 6 | Global_Common_V2_1 | 6 |
| BGB1234_101_V3 | Sex | 性别 | 6 | Global_Common_V2_1 | 6 |
| BGB1234_101_V3 | Male | 男 | 6 | Global_Common_V2_1 | 6 |
| BGB1234_101_V3 | Female | 女 | 6 | Global_Common_V2_1 | 6 |
| BGB1234_101_V3 | Is subject of child-bearing potential? | 受试者是否有生育能力? | 6 | BGB1234_101_V3 | 6 |
| BGB1234_101_V3 | Yes | 是 | 6 | Global_Common_V2_1 | 4 |
| BGB1234_101_V3 | No | 否 | 6 | Global_Common_V2_1 | 4 |
| BGB1234_101_V3 | Ethnicity | 种族 | 6 | Global_Common_V2_1 | 6 |

## CREATE XFDF FILE

We create translation annotations XFDF file with translation dictionary. Annotation contents are Chinese translated strings. Rectangular box border color and background color are opaque white. Each annotation box's coordinates are exactly same as its corresponding English string. Note: Each CRF page's origin of coordinate (0, 0) is at the left bottom conner in XFDF, which is different from English strings. We can do coordinate transformation between XFDF and English strings with PDF page height = 792 points. Y in English string = 792 – Y in XFDF. See PDF Page Coordinates for details. See Ilias Pyrnokokis, 2015 for details how to create XFDF file. Import XFDF file into the CRF file. Every English string is covered by its corresponding Chinese translated string perfectly. Only Chinese translated strings are visible. English strings are hidden underneath. It is just like a perfect mask covers a face. So, it is called the CRF translation approach as Mask Plan.

After importing XFDF into CRF, print CRF with PDF printer. New printed PDF is Chinese translated CRF. Thank to powerful Adobe, Chinese strings are searchable in new printed PDF. The last step is to translate bookmarks. We will show how to translate form names in bookmarks.

## EXTRACT ENGLISH BOOKMARKS FROM ENGLISH CRF

The approach is very similar with extracting each English letter and its coordinates from CRF. We still use Apache® PDFBox Java code in SAS® Procedure Groovy. Here are instructions how to extract English bookmarks from CRF based on our practice. You could follow it blindly without worrying about SAS® Procedure Groovy or Apache® PDFBox Java code anymore if you are not familiar with them at all.

1. Download pdfbox-app-2.0.24.jar and pdfbox-2.0.24-src.zip from https://pdfbox.apache.org/download.html.
2. Unzip pdfbox-2.0.24-src.zip to Folder pdfbox-2.0.24-src.
3. Update below Macro `m_PrintBookmarks.`
    a. Two Yellow highlighted `YourWorkingFolderPath` should be updated into your working folder path, which pdfbox-app-2.0.24.jar and Folder pdfbox-2.0.24-src are located.
    b. Assign Macro argument enPDF with your English CRF folder path and file name.
    c. Assign Macro argument enBookmarks with your defined output SAS dataset name.

```
%macro m_PrintBookmarks(enPDF=, enBookmarks=);

filename javabm "YourWorkingFolderPath/pdfbox-2.0.24-src/pdfbox-
2.0.24/examples/src/main/java/org/apache/pdfbox/examples/pdmodel/PrintB
ookmarks.java";
```

```
data PrintBookmarks;
  length STRING $2000;
  retain OutFlag InsertFlag InsertLocation;
  infile javabm length=l lrecl=2000 end=eof;
  input STRING $varying2000.l;

  if _N_=1 then
    do;
      STRING='proc groovy CLASSPATH="'||'YourWorkingFolderPath /pdfbox-
app-2.0.24.jar'||'";';
      output;
      STRING='  submit;';
      output;
    end;

  if cats(String)="import java.io.File;" then
    OutFlag=1;
  if OutFlag=1 then
    output;

  InsertLocation+1;
  if cats(String)="public static void main( String[] args ) throws
IOException" then
    do;
      InsertFlag=1;
      InsertLocation=0;
    end;
  if InsertFlag=1 and InsertLocation=1 then
    do;
      String="      args = new String[1];";
      output;
      String='      args[0] = "'||"&enPDF"||'";';
      output;
    end;

  if eof then
    do;
      String="  endsubmit;";
      output;
      String="quit;";
      output;
    end;

run;

data _null_;
  set PrintBookmarks;
  FILE "%sysfunc(pathname(work))/PrintBookmarks.sas" old;
  l=length(String);
  put STRING $varying2000.l;
run;

filename bookmark "%sysfunc(pathname(work))/PrintBookmarks.txt";
proc printto log=bookmark new;
run;
```

```
%include "%sysfunc(pathname(work))/PrintBookmarks.sas";

proc printto;
run;

data &enBookmarks;
  length STRING $2000
  ;
  infile bookmark length=l lrecl=2000 end=eof;
  input STRING $varying2000.l;
run;
%mend m_PrintBookmarks;
%m_PrintBookmarks(enPDF=YourEnglishCRFFilePathandFileName.pdf,
enBookmarks= YourDefinedOutputDatasetName)
```

Based on `%m_PrintBookmarks` output dataset defined in macro argument enBookmarks, and translation dictionary, we can derive bookmark translation dataset as below Table 3. Column Form are English Bookmarks. Column FormCN are their Chinese translations. Column Page are bookmarks linked destination.

**Table 3 Sample of Bookmarks Translation Dataset**

| Form | Page | FormCN |
|---|---|---|
| V1.0 06Apr21 1336 DEV | 1 | |
| Subject | 2 | 受试者信息 |
| Date of Visit | 4 | 访视日期 |
| Informed Consent | 6 | 知情同意书 |
| Demographics | 8 | 人口统计 |

## CREATE CHINESE BOOKMARKS INTO CHINESE TRANSLATED CRF

The approach is very similar with extracting English Bookmarks form English CRF. We still use Apache® PDFBox Java code in SAS® Procedure Groovy. Here are instructions how to create Chinese bookmarks into Chinese translated CRF based on our practice. You could follow it blindly without worrying about SAS® Procedure Groovy or Apache® PDFBox Java code anymore if you are not familiar with them at all.

1. Download pdfbox-app-2.0.24.jar and pdfbox-2.0.24-src.zip from https://pdfbox.apache.org/download.html.
2. Unzip pdfbox-2.0.24-src.zip to Folder pdfbox-2.0.24-src.
3. Update below Macro `m_CreateBookmarks.`
    a. Two Yellow highlighted `YourWorkingFolderPath` should be updated into your working folder path, which pdfbox-app-2.0.24.jar and Folder pdfbox-2.0.24-src are located.
    b. Assign Macro argument BookmarksTransDict with your Bookmarks translation dataset name. Data structure and column names follow Table 3.
    c. Assign Macro argument `cnCRF` with Chinese printed CRF file path and file name.
    d. Assign Macro argument `OutcnCRFwithBookmarks` with your given final Chinese translated CRF file path and file name, in which Chinese translated bookmarks will be created.

```
%macro m_CreateBookmarks(BookmarksTransDict=, cnCRF=,
OutcnCRFwithBookmarks=);
data bookmarks_1;
  length JavaCode $2000;
  retain FormminPageFlag ' ';
  set &BookmarksTransDict;
  if missing(FormCN) then
    FormCN=Form;
  if Page=1 then
    do;
      FirstPageForm="Y";
      JavaCode='
pagesOutline.setTitle( "'||strip(FormCN)||'" );';
      output;
    end;
  else
    do;
      FormBookMarkFlag="Y";
      if FormminPageFlag=" " then
        do;
          javaCode='                    if( pageNum == '||cats(Page)||' )';
          FormminPageFlag="Y";
        end;
      else
        do;
          javaCode='                    else if( pageNum ==
'||cats(Page)||' )';
        end;
      output;
      javaCode='                  {';
      output;
      javaCode='                    PDOutlineItem bookmark = new
PDOutlineItem();';
      output;
      javaCode='                    bookmark.setDestination( dest );';
      output;
      javaCode='
bookmark.setTitle( "'||strip(FormCN)||'");';
      output;
      javaCode='                    pagesOutline.addLast( bookmark );';
      output;
      javaCode='                  }';
      output;
    end;


run;



filename javabm "YourWorkingFolderPath/pdfbox-2.0.24-src/pdfbox-
2.0.24/examples/src/main/java/org/apache/pdfbox/examples/pdmodel/CreateBookma
rks.java";


data CreateBookmarks;
```

```
  length STRING LastString $2000;
  retain OutFlag InsertFlag InsertLocation FormBookMarkFlag;
  infile javabm length=l lrecl=2000 end=eof;
  input STRING $varying2000.l;

  LastString=lag(String);

  if _N_=1 then
    do;
      STRING='proc groovy CLASSPATH="'||'YourWorkingFolderPath/pdfbox-app-
2.0.24.jar'||'";';
      output;
      STRING='  submit;';
      output;
    end;

  if strip(String)='private CreateBookmarks()' then
    do;
      String="    public CreateBookmarks()";
    end;
  else if strip(String)='pagesOutline.setTitle( "All Pages" );' then
    do;
      FirstPageForm="Y";
    end;
  else if strip(String)='PDOutlineItem bookmark = new PDOutlineItem();'  then
    do;
      FormBookMarkFlag="Y";
      OutFlag=0;
    end;
  else if strip(String)='pagesOutline.addLast( bookmark );' then
    do;
      OutFlag=1;;
    end;
  else if strip(LastString)='pagesOutline.addLast( bookmark );' then
    do;
      FormBookMarkFlag=" ";
    end;

  if cats(String)="import java.io.File;" then
    OutFlag=1;

  InsertLocation+1;
  if strip(String)="public static void main( String[] args ) throws
IOException" then
    do;
      InsertFlag=1;
      InsertLocation=0;
    end;


  if InsertFlag=1 and InsertLocation=1 then
    do;
      output;
      String="      args = new String[2];";
      output;
      String='      args[0] = "'||"&cnCRF"||'";';
      output;
```

```
      String='      args[1] = "'||"&OutcnCRFwithBookmarks"||'";';
      output;
    end;
  else if OutFlag=1 then
    output;

  if eof then
    do;
      String="  endsubmit;";
      output;
      String="quit;";
      output;
    end;

run;

data CreateBookmarks_1;
  set CreateBookmarks;
  SEQ+1;
run;

data bookmarks_2;
  set bookmarks_1;
  SEQ+1;
run;

proc sql;
  create table CreateBookmarks_2 as
    select coalescec(b.JavaCode, a.String) as String, a.FirstPageForm,
a.FormBookMarkFlag
      from CreateBookmarks_1 as a
        left join bookmarks_2 as b
        on a.FirstPageForm=b.FirstPageForm
          and a.FormBookMarkFlag=b.FormBookMarkFlag
      order by a.SEQ, b.SEQ;
quit;

data _null_;
  set CreateBookmarks_2;
  FILE "%sysfunc(pathname(work))/CreateBookmarks.sas" old;
  l=length(String);
  put STRING $varying2000.l;
run;

%include "%sysfunc(pathname(work))/CreateBookmarks.sas";
%mend m_CreateBookmarks;

%m_CreateBookmarks(BookmarksTransDict= BookmarksTranslationDataset,
cnCRF=ChineseTranslatedCRFwithoutBookmarks
, OutcnCRFwithBookmarks= ChineseTranslatedCRFwithBookmarks)
```

## AUTO-TRANSLATION THROUGH TRANSLATION API

There are so many Translation APIs, which allow us to incorporate machine translation into CRF translation. See Top 10 Best Translation APIs in 2021 for details. For example, you can use following SAS code to use free Google translate API for test. See How to Use the Google Translate API for Free for details.

```
%let StringEN=%sysfunc(urlencode(Hello World));

filename tempf url
"%nrstr(https://translate.googleapis.com/translate_a/single?client=gtx&
sl=en&tl=zh-CN&dt=t&q=)%nrbquote(&StringEN)" debug LRECL=32767;

data temp;
  infile tempf lrecl=2000 truncover;
  input String $ 1-2000;
run;
```

Since it is free, there is some quota limitation. If your translation requests are more some number of requests per day, you cannot access the free Google translate API anymore from your IP address that day.

If you get your authentication keys from Azure Translator, you can also use Azure translator API in SAS as below code with yellow highlighted authentication keys.

```
%let InString="[{'text': '%nrbquote(Hello World)'}]";
filename resp TEMP;
proc http URL='https://api.cognitive.microsofttranslator.com/translate?api-
version=3.0&from=en&to=zh-Hans' out=resp method="post" in=&InString;
 HEADERS
  'Ocp-Apim-Subscription-Key'='xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
  'Ocp-Apim-Subscription-Region'='global'
  'Content-type'='application/json'
  "charset"="UTF-8"
;
run;
data temp;
  infile resp lrecl=2000 truncover;
  input String $ 1-2000;
run;
```

## CONCLUSION

Through Mask Plan, in a new trial, estimated more than 80% CRF English strings' Chinese translations can be directly copied from previous trials. The more trials' CRFs are translated, the more translation work will be saved in new trials. One English string is translated only once. And its Chinese translation is consistent across trials. Translated Chinese strings are located at exactly same location as their corresponding English strings. When SDTM annotations are imported from English SDTM aCRF, we don't need to adjust their location, because CRF English strings are translated into Chinese in place. And bookmarks are also translated. With Mask Plan, we can focus on translation dictionary, all others are automatic. And with translation APIs, only a little manual review work is needed on new English strings' Chinese translations.

## REFERENCES


Professor Norman, 2014. "Lucy Quotes" https://www.magicalquote.com/moviequotes/the-sole-purpose-of-life/

Yin-Jhen Yan etc., 2018 "No Solution to Auto-generating acrf.pdf? Try to Use GROOVY Procedure!" https://www.pharmasug.org/proceedings/china2018/AD/Pharmasug-China-2018-AD71.pdf

Using PDFBox to locate text coordinates within a PDF in Java | jackson-brain.com

PDF Page Coordinates, https://www.pdfscripting.com/public/PDF-Page-Coordinates.cfm

Top 10 Best Translation APIs in 2021 https://blog.api.rakuten.net/top-10-best-translation-apis-google-translate-microsoft-translator-and-others/

How to Use the Google Translate API for Free https://www.labnol.org/code/19909-google-translate-api

## CONTACT INFORMATION

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

Chunpeng Zhao
BeiGene
Chunpeng.zhao@beigene.com
Liming Xie
BeiGene
liming.xie@beigene.com
Chunling Xu
BeiGene
chunling.xu@beigene.com