PharmaSUG 2025 - Paper AP-037

Listing Shell to SAS® Program Automation Tool

Balaji Ayyappan, ICON Plc;
Johnny Tai, Kite Pharma

ABSTRACT

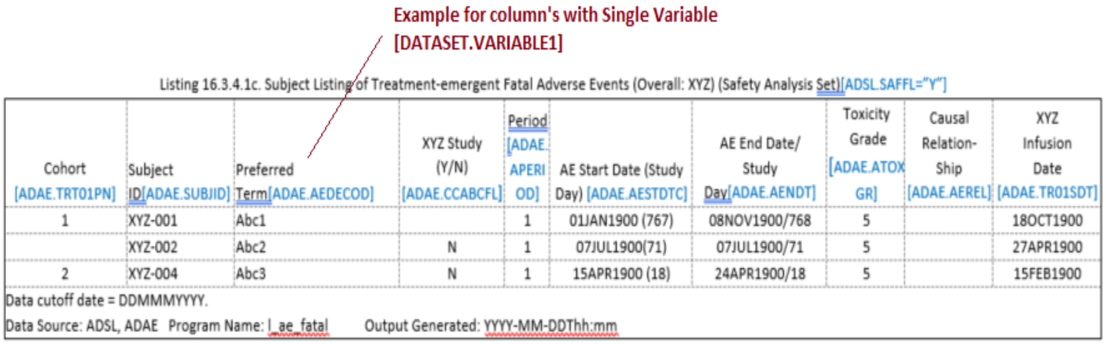
Listings in clinical reports are required part of the documentation for clinical trials and used for statistical review by regulatory authority. We created a tool that can automatically generate the SAS® programs to produce multiple listings, when we pass the TFL shell document (Table, Figures and Listings shells). Tool can identify the listing shell pages from shell and uses the shell annotation information, metadata and value level information from the datasets while generating the SAS code. SAS and VBA applications are used to develop this tool. Based on the annotated variables, for example: analysis flag used in title – we can subset required subjects from analysis dataset, character or numeric date variables are converted to company specified character date format in reporting, columns width in proc report procedure and page number variable to determine number of lines to be printed in each page are automatically determined. If any column reporting two or more variables, tool derives the concatenated variable based on the symbol present in column header. If Y/N values present in dataset and in Shell if it has Yes/No, tool read both dataset and shell info and automatically generate formats. If any column required customized coding that can also be included in the code using key words. This tool can evolve, we can add more features to it and this tool is extended to produce standard Tables and Figures.

INTRODUCTION

We use VBA-userform in Word application to get shell document location from the user and using SAS macro to create excel file with metadata and value level information from datasets. VBA macros process through annotated Listing shell document, based on annotated variables, it takes the corresponding metadata information (like variable char/num type, label, etc.) and value level information (like sample values, variable value's max and min length, no. of unique values, date formats are determined from the sample values, etc.) to create Listing SAS program. This tool can also be used to create a Single listing program. We follow below steps to run this tool.

STEP1: We run the SAS Macro to get the metadata and value level information from the datasets into excel file. We annotate the listing shells, please refer below types of specifications of annotated Listing Shell.

Example Listing Shell-1: Columns which are using single variable, use square bracket with [DATASET.VARIABLE]. Please check below example.



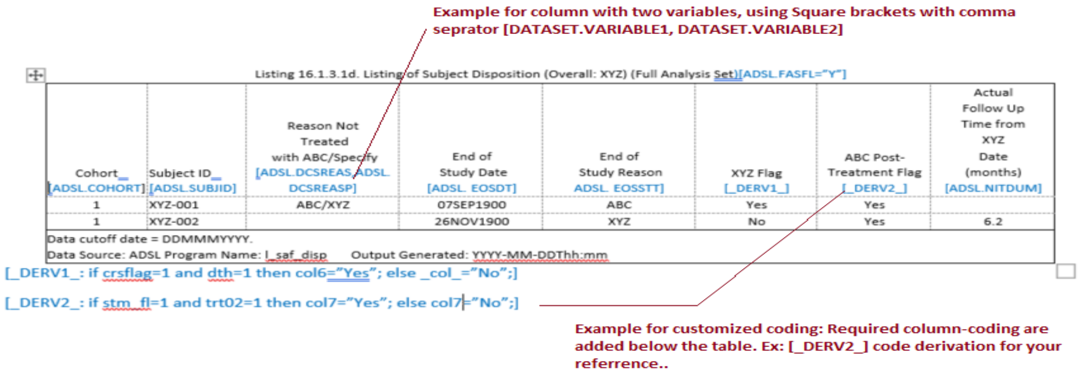
Example for column's with Single Variable
[DATASET.VARIABLE1]

Listing 16.3.4.1c. Subject Listing of Treatment-emergent Fatal Adverse Events (Overall: XYZ) (Safety Analysis Set)[ADSL.SAFFL="Y"]

Cohort [ADAE.TRT01PN]	Subject ID[ADAE.SUBJID]	Preferred Term[ADAE.AEDECOD]	XYZ Study (Y/N) [ADAE.CCABCFL]	Period [ADAE.APERIOD]	AE Start Date (Study Day) [ADAE.AESTDTC]	AE End Date/ Study Day[ADAE.AENDT]	Toxicity Grade [ADAE.ATOXGR]	Causal Relation-Ship [ADAE.AEREL]	XYZ Infusion Date [ADAE.TR01SDT]
1	XYZ-001	Abc1		1	01JAN1900 (767)	08NOV1900/768	5		18OCT1900
	XYZ-002	Abc2	N	1	07JUL1900(71)	07JUL1900/71	5		27APR1900
2	XYZ-004	Abc3	N	1	15APR1900 (18)	24APR1900/18	5		15FEB1900

Data cutoff date = DDMMMYYYY.
Data Source: ADSL, ADAE   Program Name: l_ae_fatal   Output Generated: YYYY-MM-DDThh:mm

Example Listing Shell-2: Columns using multiple variables should use square bracket, variables separated by commas [DATASET.VARIABLE1, DATASET.VARIABLE2]. If any column required more coding for deriving, customized coding can be added to listing program using keyword _DERV1_, _DERV2_, etc.., [Ex: _DERV1_ column customize code in the below shell.]



Example for column with two variables, using Square brackets with comma seprator [DATASET.VARIABLE1, DATASET.VARIABLE2]

Listing 16.1.3.1d. Listing of Subject Disposition (Overall: XYZ) (Full Analysis Set)[ADSL.FASFL="Y"]

Cohort_ [ADSL.COHORT]	Subject ID_ [ADSL.SUBJID]	Reason Not Treated with ABC/Specify [ADSL.DCSREAS,ADSL. DCSREASP]	End of Study Date [ADSL. EOSDT]	End of Study Reason ADSL. EOSSTT]	XYZ Flag [_DERV1_]	ABC Post-Treatment Flag [_DERV2_]	Actual Follow Up Time from XYZ Date (months) [ADSL.NITDUM]
1	XYZ-001	ABC/XYZ	07SEP1900	ABC	Yes	Yes	
1	XYZ-002		26NOV1900	XYZ	No	Yes	6.2

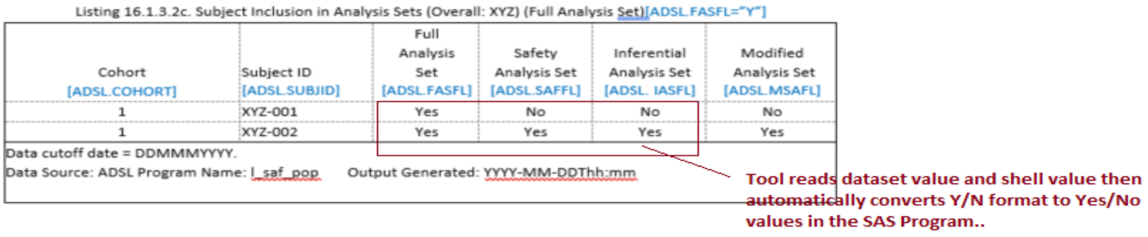
Data cutoff date = DDMMMYYYY.
Data Source: ADSL Program Name: l_saf_disp Output Generated: YYYY-MM-DDThh:mm

[_DERV1_: if crsflag=1 and dth=1 then col6="Yes"; else _col_="No";]

[_DERV2_: if stm_fl=1 and trt02=1 then col7="Yes"; else col7="No";]

Example for customized coding: Required column-coding are added below the table. Ex: [_DERV2_] code derivation for your referrence..

Example Listing Shell-3: Suppose in Listing output you want to print Yes/No values, but when you have Y/N values in dataset, tool can read dataset value and Shell value, automatically creates the Yes/No format in the listing SAS program.



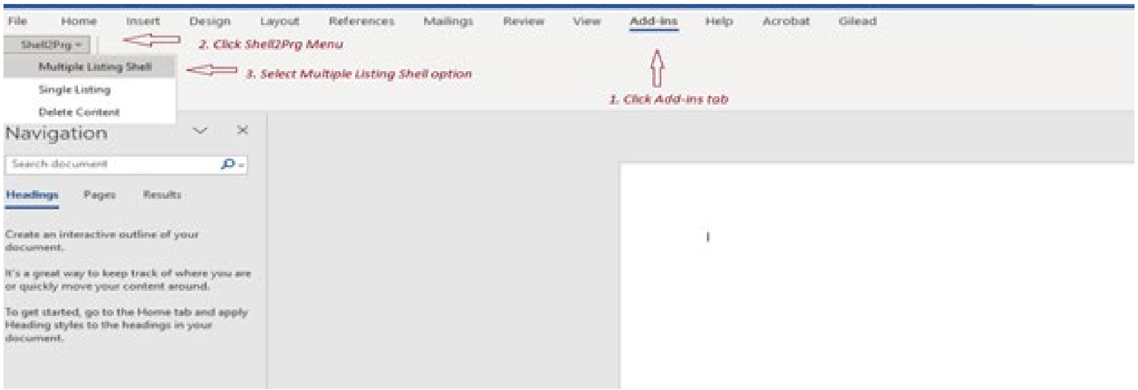
Listing 16.1.3.2c. Subject Inclusion in Analysis Sets (Overall: XYZ) (Full Analysis Set)[ADSL.FASFL="Y"]

Cohort [ADSL.COHORT]	Subject ID [ADSL.SUBJID]	Full Analysis Set [ADSL.FASFL]	Safety Analysis Set [ADSL.SAFFL]	Inferential Analysis Set [ADSL. IASFL]	Modified Analysis Set [ADSL.MSAFL]
1	XYZ-001	Yes	No	No	No
1	XYZ-002	Yes	Yes	Yes	Yes

Data cutoff date = DDMMMYYYY.
Data Source: ADSL Program Name: l_saf_pop Output Generated: YYYY-MM-DDThh:mm

Tool reads dataset value and shell value then automatically converts Y/N format to Yes/No values in the SAS Program..

STEP 2: Using word vba application, we click on the 'Add-ins" menu, it will show "Shell2Prg" dropdown menu, then select "Multiple Listing Shell" as shown below.



2. Click Shell2Prg Menu

3. Select Multiple Listing Shell option

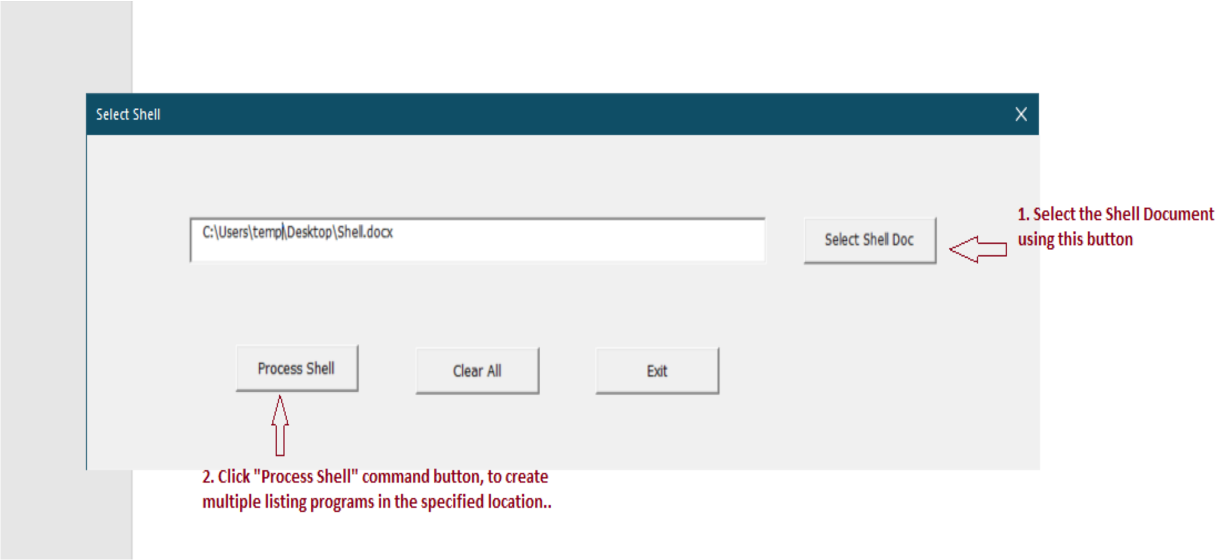
1. Click Add-ins tab

Create an interactive outline of your document.

It's a great way to keep track of where you are or quickly move your content around.

To get started, go to the Home tab and apply Heading styles to the headings in your document.

STEP3: Below userform will be opened, user can select the annotated Listing Shell document which contains multiple listing shell by clicking “Select Shell Doc” command button. Then click the “Process Shell” command button, it will process the shell document and create the listing programs in specified locations with respect to annotation information. Please check the below snapshot.

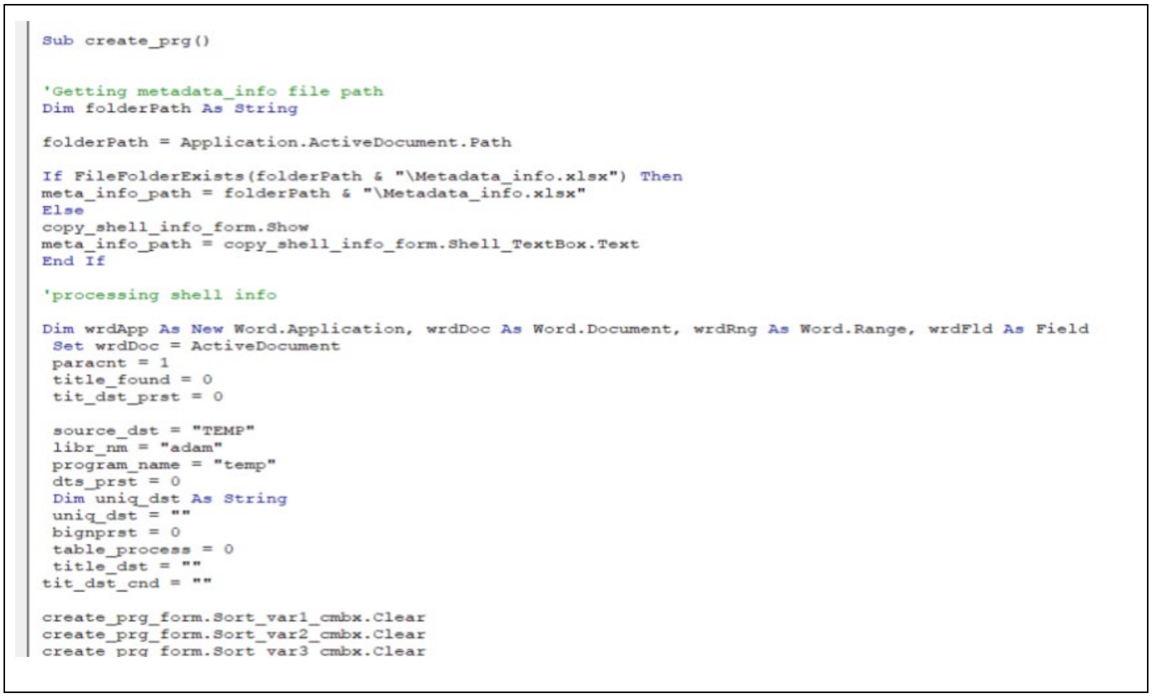


C:\Users\temp\Desktop\Shell.docx

1. Select the Shell Document using this button

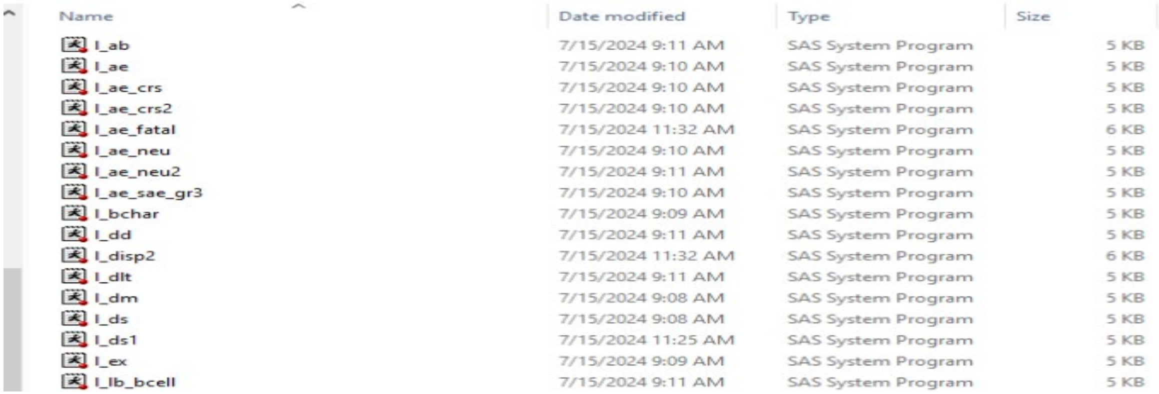
2. Click "Process Shell" command button, to create multiple listing programs in the specified location..

STEP 4: Snapshot of VBA code used in the tool.



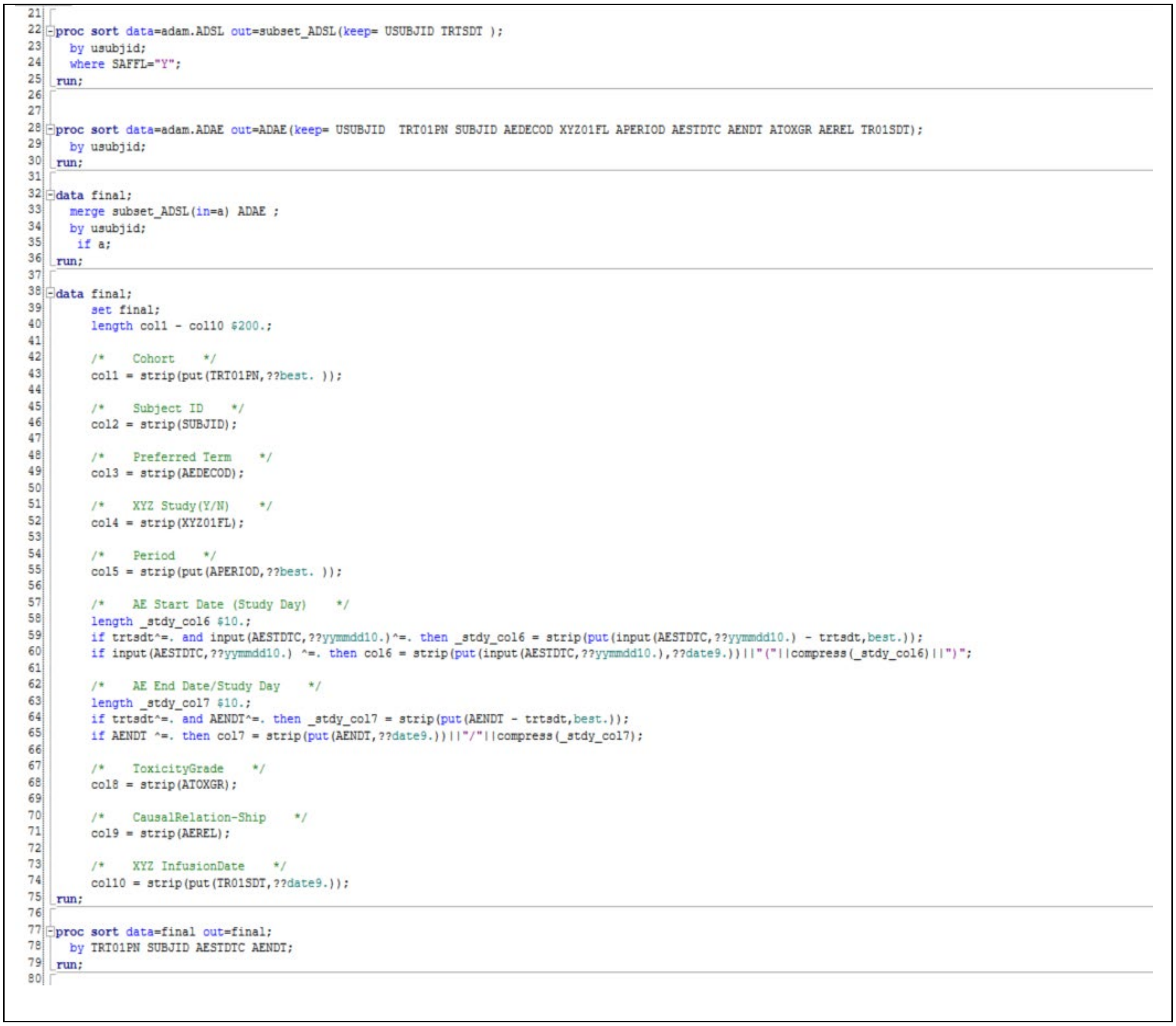
```
Sub create_prg()

'Getting metadata_info file path
Dim folderPath As String

folderPath = Application.ActiveDocument.Path

If FileFolderExists(folderPath & "\Metadata_info.xlsx") Then
meta_info_path = folderPath & "\Metadata_info.xlsx"
Else
copy_shell_info_form.Show
meta_info_path = copy_shell_info_form.Shell_TextBox.Text
End If

'processing shell info

Dim wrdApp As New Word.Application, wrdDoc As Word.Document, wrdRng As Word.Range, wrdFld As Field
 Set wrdDoc = ActiveDocument
 paracnt = 1
 title_found = 0
 tit_dst_prst = 0

 source_dst = "TEMP"
 libr_nm = "adam"
 program_name = "temp"
 dts_prst = 0
 Dim uniq_dst As String
 uniq_dst = ""
 bignprst = 0
 table_process = 0
 title_dst = ""
tit_dst_cnd = ""

create_prg_form.Sort_var1_cmbx.Clear
create_prg_form.Sort_var2_cmbx.Clear
create_prg_form.Sort_var3_cmbx.Clear
```

STEP 5: Listing programs are created automatically in the specified location.



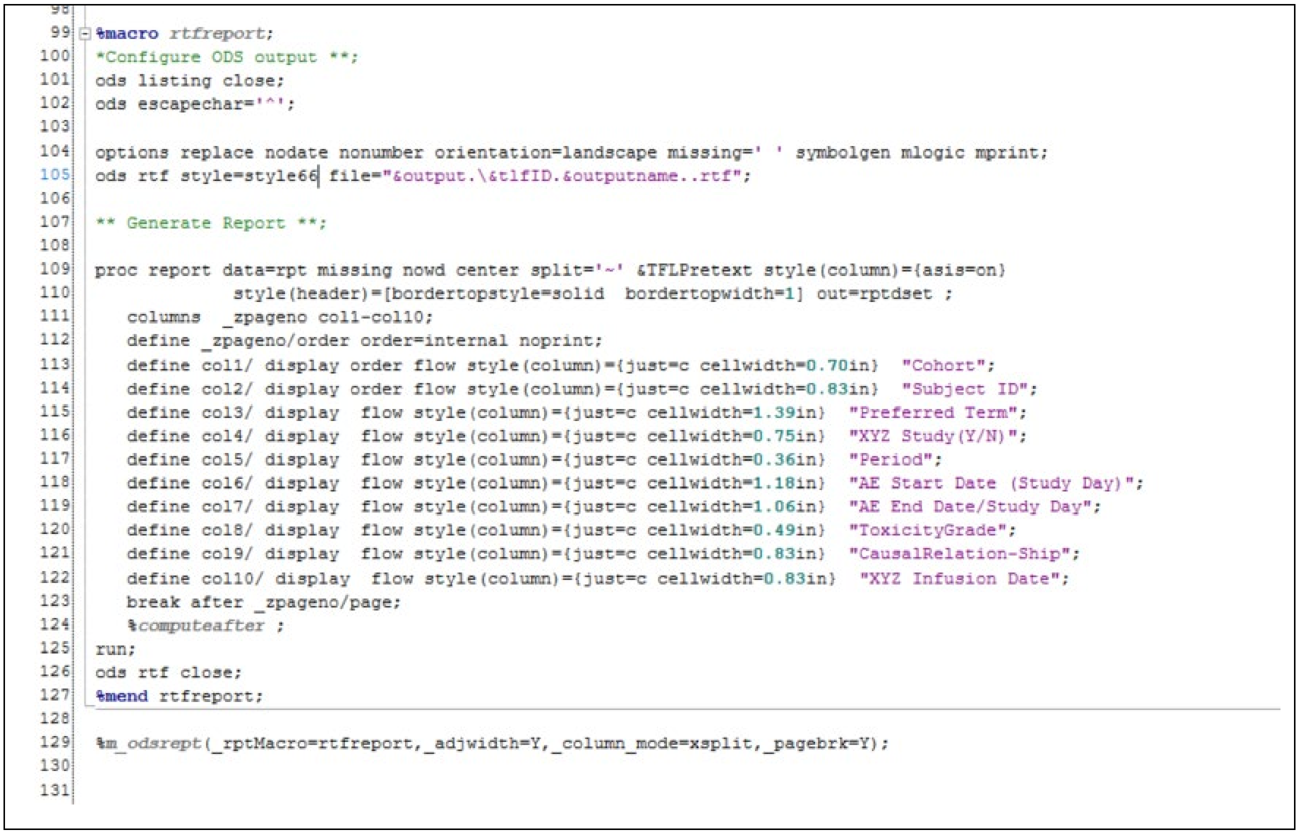
Name	Date modified	Type	Size
l_ab	7/15/2024 9:11 AM	SAS System Program	5 KB
l_ae	7/15/2024 9:10 AM	SAS System Program	5 KB
l_ae_crs	7/15/2024 9:10 AM	SAS System Program	5 KB
l_ae_crs2	7/15/2024 9:10 AM	SAS System Program	5 KB
l_ae_fatal	7/15/2024 11:32 AM	SAS System Program	6 KB
l_ae_neu	7/15/2024 9:10 AM	SAS System Program	5 KB
l_ae_neu2	7/15/2024 9:11 AM	SAS System Program	5 KB
l_ae_sae_gr3	7/15/2024 9:10 AM	SAS System Program	5 KB
l_bchar	7/15/2024 9:09 AM	SAS System Program	5 KB
l_dd	7/15/2024 9:11 AM	SAS System Program	5 KB
l_disp2	7/15/2024 11:32 AM	SAS System Program	6 KB
l_dlt	7/15/2024 9:11 AM	SAS System Program	5 KB
l_dm	7/15/2024 9:08 AM	SAS System Program	5 KB
l_ds	7/15/2024 9:08 AM	SAS System Program	5 KB
l_ds1	7/15/2024 11:25 AM	SAS System Program	5 KB
l_ex	7/15/2024 9:09 AM	SAS System Program	5 KB
l_lb_bcell	7/15/2024 9:11 AM	SAS System Program	5 KB

STEP 6: Snapshot of SAS code created in sample listing program. SAS code will sort listing by treatment and subject variables, dates are also used in sort order for certain listings (example: adverse listing). Validation dataset is created in specified location.



```
proc sort data=adam.ADSL out=subset_ADSL(keep= USUBJID TRTSDT );
  by usubjid;
  where SAFFL="Y";
run;


proc sort data=adam.ADAE out=ADAE(keep= USUBJID  TRT01PN SUBJID AEDECOD XYZ01FL APERIOD AESTDTC AENDT ATOXGR AEREL TR01SDT);
  by usubjid;
run;

data final;
  merge subset_ADSL(in=a) ADAE ;
  by usubjid;
   if a;
run;

data final;
     set final;
     length col1 - col10 $200.;

     /*    Cohort    */
     col1 = strip(put(TRT01PN,??best. ));

     /*    Subject ID    */
     col2 = strip(SUBJID);

     /*    Preferred Term    */
     col3 = strip(AEDECOD);

     /*    XYZ Study(Y/N)    */
     col4 = strip(XYZ01FL);

     /*    Period    */
     col5 = strip(put(APERIOD,??best. ));

     /*    AE Start Date (Study Day)    */
     length _stdy_col6 $10.;
     if trtsdt^=. and input(AESTDTC,??yymmdd10.)^=. then _stdy_col6 = strip(put(input(AESTDTC,??yymmdd10.) - trtsdt,best.));
     if input(AESTDTC,??yymmdd10.) ^=. then col6 = strip(put(input(AESTDTC,??yymmdd10.),??date9.))||"("||compress(_stdy_col6)||")";

     /*    AE End Date/Study Day    */
     length _stdy_col7 $10.;
     if trtsdt^=. and AENDT^=. then _stdy_col7 = strip(put(AENDT - trtsdt,best.));
     if AENDT ^=. then col7 = strip(put(AENDT,??date9.))||"/"||compress(_stdy_col7);

     /*    ToxicityGrade    */
     col8 = strip(ATOXGR);

     /*    CausalRelation-Ship    */
     col9 = strip(AEREL);

     /*    XYZ InfusionDate    */
     col10 = strip(put(TR01SDT,??date9.));
run;

proc sort data=final out=final;
  by TRT01PN SUBJID AESTDTC AENDT;
run;
```

STEP 7: Initially column width used in the shell document are processed through vba and used in the proc report procedure code within the rtfreport macro which is processed by the odsreport macro. Proc stream procedure store the initial proc report statement in rtfreport macro. These initial statements served as specifications for subsequent column width analysis. The analysis goal is to prevent word-split and page overflow in the final table. Word-split is controlled by ensuring column width is no less than the maximum word width for a column variable based on the actual data value and font attributes. With all calculated column widths, table, and page layout information, we check for page overflow. If all checks passed, the stored proc report statements are then updated and run to create the table.



```
%macro rtfreport;
*Configure ODS output **;
ods listing close;
ods escapechar='^';

options replace nodate nonumber orientation=landscape missing=' ' symbolgen mlogic mprint;
ods rtf style=style66 file="&output.\&tlfID.&outputname..rtf";

** Generate Report **;

proc report data=rpt missing nowd center split='~' &TFLPretext style(column)={asis=on}
             style(header)=[bordertopstyle=solid  bordertopwidth=1] out=rptdset ;
   columns  _zpageno col1-col10;
   define _zpageno/order order=internal noprint;
   define col1/ display order flow style(column)={just=c cellwidth=0.70in}  "Cohort";
   define col2/ display order flow style(column)={just=c cellwidth=0.83in}  "Subject ID";
   define col3/ display  flow style(column)={just=c cellwidth=1.39in}  "Preferred Term";
   define col4/ display  flow style(column)={just=c cellwidth=0.75in}  "XYZ Study(Y/N)";
   define col5/ display  flow style(column)={just=c cellwidth=0.36in}  "Period";
   define col6/ display  flow style(column)={just=c cellwidth=1.18in}  "AE Start Date (Study Day)";
   define col7/ display  flow style(column)={just=c cellwidth=1.06in}  "AE End Date/Study Day";
   define col8/ display  flow style(column)={just=c cellwidth=0.49in}  "ToxicityGrade";
   define col9/ display  flow style(column)={just=c cellwidth=0.83in}  "CausalRelation-Ship";
   define col10/ display  flow style(column)={just=c cellwidth=0.83in}  "XYZ Infusion Date";
   break after _zpageno/page;
   %computeafter ;
run;
ods rtf close;
%mend rtfreport;

%m_odsrept(_rptMacro=rtfreport,_adjwidth=Y,_column_mode=xsplit,_pagebrk=Y);
```

We have following limitations using this tool: Tool cannot be used when dataset/variable information is in vertical structure and need to be transposed to create the listing. Tool cannot be used, when merging two or more datasets with selected variables to create the listings. Currently Tool can manage simple merge from two or more datasets using 'usubjid.'

CONCLUSION

Using this we can create multiple listings in few minutes, whereas if we create manually, it requires more time and more resources. For example: If we manually create one Listing it takes around 2-3 hrs. for an experienced programmer, this tool can create 30-40 listings in 5-10 mins. In most cases, listing SAS programs that are created are already first draft ready. This tool also checks the annotated variable name against the metadata, provides the notes for non-existing variables. We can add more features to this tool and make listing generations much easier. We have extended this tool to create the SAS programs for standard tables and graphs.

ACKNOWLEDGMENTS

We like to thank Jessie Teng (Kite Pharma) for all the support and guidance provided for developing this tool.

CONTACT INFORMATION

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

Balaji Ayyappan,
ICON Plc,
balaji.ayyappan@iconplc.com

Johnny Tai,
Kite Pharma,
johnny.tai1@gilead.com