

## Macro Makes Final Documentation Quick and Easy

Indrani Sarkar, inVentiv Health Clinical, Indianapolis, Indiana, USA

Jean Crain, inVentiv Health Clinical, Indianapolis, Indiana, USA

### ABSTRACT

Purpose of this macro is to generate a document containing information on all the sas programs used to create derived datasets, summary, listings, and figures during a clinical trial project. This word document is used as a part of final package delivered to the client after completion of each project. Primary objective for this file is to assemble all the key information in one document so that it would be easier to replicate all the work in future without any difficulties and also it will help someone to get familiar with a project in a very short time. The macro, explained in this paper, will be able to list the sas program names; the analysis population(s); the endpoints; name(s) of required input datasets (including non-sas files); any additional SAS macros called by the program; the output file name(s); and any additional notes clarifying information. It would also be able to group tables, listings and figures based on their endpoints. Macro uses PRVF (Programming Review and Validation form), an excel workbook which keeps track of deliveries made at different points of time, and all programs as source files.

Our expectation is that this macro will reduce number of hours usually spent on the documentation as well as human errors and would bring consistency in format across different projects.

### INTRODUCTION

Clinical trial projects usually go on for months, sometimes years, and include several draft deliveries before a final one. At the end of this long process, it becomes absolutely necessary to have proper documentation of all the analyses done at different time points. Usually a final package containing all the study related documents like SAP, specs, programs, logs, macros and a Word® file called FDA doc. FDA Doc is created to serve two purposes: (1) to help someone to get familiar with the project within little time and (2) to send it as a part of submission package to FDA. FDA doc contains key information like program names for derived datasets and TLFs (Tables, Listings and Figures), population analyzed, Endpoints, Inputs (including macros used), and outputs. The expectation is to complete the whole package within a week. Therefore the FDA doc needs to be done in very short time and frequently by someone who is not with the project from its inception. It usually takes at least two working days to create a high quality file. Having worked on several final docs within a few weeks, we felt the need of a good script which will create the file with those key information in a very short time (may be within few minutes). This will give more time to pay attention to details and will also minimize human errors. This should also help the reviewer to have confidence in correctness of this part of the package. Below is the image of a sample final document.

SAS v9 Program Name	Population Analyzed	Endpoint Analyzed	Input /Output SAS xport file(s) (if applicable)	Output
<b>General Programs</b>				
ae.sas	N/A	N/A	<b>Input:</b> adv.sas7bdat, dov.sas7bdat, popula.sas7bdat. <b>Included Files:</b> cleanup.sas, getlabel.sas.	ae.sas7bdat
<b>Conduct of Study</b>				
cdl_demo.sas	Safety	Demographic Characteristics	<b>Input:</b> ic.sas7bdat, popula.sas7bdat <b>Included Files:</b> cleanup.sas, headfoot.sas,.	16.2.4.1_cdl_demo.htm
<b>Safety</b>				
cdl_ae.sas	Safety	Adverse Events	<b>Input:</b> ae.sas7bdat, hcu.sas7bdat, popula.sas7bdat. <b>Included Files:</b> cleanup.sas, headfoot.sas, incl.sas, logchecker.sas, readdata.sas.	16.2.7.10_cdl_ae_nonprot.htm cdl_ae cdl_ae_rel.htm

Display 1: Example of a Typical FDA Doc.

## EXISTING PROCESS:

The current process of making FDA Doc is very time intensive and requires reviewing of several project related documents, like Programming Review and Validation Form (henceforth PRVF), Statistical Analysis Plan (SAP) etc, simultaneously. PRVF, an excel spreadsheet, contains all the names of programs and corresponding outputs produced during different deliveries. All the program and output names are entered in FDA doc manually from this spreadsheet. SAP is used to collect the information on endpoints and analysis population. So a good amount of time needs to be spent to go through both SAP and PRVF. Another big challenge is to get the names of all input datasets used in each and every program. It is expected that the program should have read all the input files/datasets at the beginning. Unfortunately, this doesn't happen in practice. Therefore, the creator of FDA Doc needs to go through each and every line in a program to find all the source datasets/files. Same thing is done for all the macros used in production programs. This is extremely time consuming task. Also, there is a high chance of omissions of a file. Given the fixed amount of time for each package, other components, like putting together programs, logs, study specific documents etc., would not get as much attention. Our expectation is that this macro will help to minimize these issues.

## METHODS

As mentioned earlier, a typical FDA Doc contains columns for program names of derived datasets & TLFs (Tables, Listing and Figures), population analyzed, endpoint analyzed, Inputs (SAS datasets and macros), and output file names. This macro reads PRVF for the validated (performed by another statistical programmer) and Qced (performed by statisticians) program and corresponding output names. It then processes each file in that directory as a text file, looking for input datasets using our libname structure: save.\* or analysis.\*. In some programs, excel or text files are also used as source of input data. It is also capable of looking for the words excel, import or infile.

```

if _N_=1 then do;
    retain libs;
    libs = prxparse('/(save.)/');
    if missing(libs) then do;
        putlog 'ERROR: regex libs is missing';
        stop;
    end;
end;

if prxmatch(libs,l_textstr) GT 0 then do;
    line = _N_;
    call prxsubstr(libs,l_textstr,libs_start,libs_len);
    start = sum (of libs_start, libs_len);
    dsn_str = substr(l_textstr,start);
    dsn = scan(dsn_str,1,' ');
    if libs_start > 0 then output;
    line = .;
end;

if indexw(l_textstr,'import') GT 0 then do;
    line = _N_;
    l_textstr = strip(translate(l_textstr,' ','=' " '));
    l_textstr = translate(l_textstr," "," ' " ");
    pos = indexw(l_textstr,'datafile');
    dsn = substr(l_textstr,pos+9);
    dsntype = scan(l_textstr,2, '.');
    if indexw(l_textstr,'excel') GT 0 then dsntype = 'xls';
    if dsn ne ' ' then output;
    line = .;
end;

```

At the same time the program looks for called macros but can exclude in-program macros which are used only to take advantage of %if / %then / %else or other macro logic and parameter use. The variable l\_textstr is a line read in from the program. The variable c\_textstr is a compressed text line of the code. Certain uses of % symbol are filtered out before the scan or substring for the macro name in the line of code takes place.

```

if (l_textstr =: '%')and
    (substr(l_textstr,1,5) ne '%let ') and
    (substr(l_textstr,1,6) ne '%macro ') and

```

```
(substr(l_textstr,1,5) not in ('%mend ','%mend;')) then do;
  line = _N_;
  macros = scan(l_textstr,1,' ');
  maclen = index(macros,' ');
  if maclen GT 0 then do;
    macros = substr(macros,1,maclen-1);
  end;
  if macros not in ('%let ','%if ','%do ','%else ','%end ','%end;','%put ')
  then do;
    macros = left(translate(macros,' ','% ;'));
    output;
  end;
end;
```

Then lists of input and macros are each made into a text string so there is one line per program. The PRVF data is processed the same way to produce one text string of output names per program and the datasets are merged, keeping programs from the PRVF data.

This macro orders all the program names based on endpoint categories. It could be done several ways but because of our naming conventions, we can do it by program name. Other options would be adding a column to the PRVF or titles-footnotes file.

```
if (pgmname=: 'saf_ae'
  or pgmname=: 'saf_sae'
  or pgmname=: 'cdl_ae'
  or pgmname=: 'cdl_sae'
  or pgmname=: 'cdl_death'
  or pgmname=: 'cdl_hosp') then do;
  cat = 'Safety';
  ord = 6;
end;
else if (pgmname=: 'saf_lr' or pgmname=: 'cdl_lr') then do;
  cat = 'Local Reactions';
  ord = 4;
end;
else if (pgmname=: 'saf_se' or pgmname=: 'cdl_se') then do;
  cat = 'Systemic Events';
  ord = 5;
end;
else if (pgmname=: 'imm') then do;
  cat = 'Immunogenicity';
  ord = 3;
end;
else do;
  cat = 'Conduct of Study';
  ord = 2;
end;
if prxmatch(assays,pgmname) GT 0
or pgmname=: 'cdl_imm' then do;
  cat = 'Immunogenicity ';
  ord = 3;
end;
if "&type" = "Analysis" then do;
  cat = 'Analysis';
  ord = 1;
end;
```

The user will only need to specify the project information at the beginning of the macro. Then it will do rest of the task. Parent directory is used in the path and the program directory PGM is expected following. PRVF and Titles-Footnotes file location need to be provided.

```
%let project=AFC1020;
%let parent=C:\Projects\1020\;
%let prvf= C:\Projects\1020\PRVF_1020.xls;
%let tf= C:\Projects\1020\titles_footnotes.xls;
```

## CONCLUSION

In the CRO world, we promise to provide best quality of work within a very short time. We are always searching for areas needing some improvement. We spend a good amount of time making the whole final package, along with the FDA doc as one piece of that package. We realized this area definitely needs to be more streamlined and could be done programmatically with review needed, rather than an author creating it. This macro significantly reduces the amount of time needed to create the FDA doc. It will need least amount of intervention and hence will minimize errors. We will be able to pay more attention to other components of the final package. It will also bring consistency across all other FDA docs. So we will know what to expect and where to look for a specific information in the document. As this macro needs minimal involvement from the creator thus anyone with little or no knowledge of the project can create it with no difficulties. Therefore we will save lot of time to get familiar with the project.

## ACKNOWLEDGMENTS

We would like to thank Fernando Enriquez of inVentiv Health Clinical for letting us use some of the code from his Lead Toolbox macros to read directories. We would also like to thank the programmers who have used this program to produce a draft Final Doc and helped to perform quality control checks and provide valuable feedback.

## CONTACT INFORMATION

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

Name: Indrani Sarkar  
Enterprise: inVentiv Health Clinical, LLP  
City, State ZIP: Indianapolis, IN  
E-mail: [Indrani.Sarkar@inventivhealth.com](mailto:Indrani.Sarkar@inventivhealth.com)

Name: Jean Crain  
Enterprise: inVentiv Health Clinical, LLP  
City, State ZIP: Indianapolis, IN  
E-mail: [Jean.Crain@inventivhealth.com](mailto:Jean.Crain@inventivhealth.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.

Other brand and product names are trademarks of their respective companies.