Paper PO02

A Simple Method to Generate Patient Profiles

Suwen Li, Cangene Corporation, Winnipeg, MB
Daniel Li, Cangene Corporation, Winnipeg, MB
Stephanie Sproule, Cangene Corporation, Winnipeg, MB
David Zhu, former Cetero Research, Toronto, ON, now i3 Statprobe, Burlington, ON

ABSTRACT

Generating patient profiles is a tedious job for Programmers at pharmaceutical companies and CROs since a lot of data manipulation and procedures, such as the PRINT and REPORT, are involved. This paper describes a simple approach to generating patient profiles that follow the order in which data is collected on the Case Report Forms (CRF). We introduce a macro that automatically creates SAS code to generate tables by using the CALL EXECUTE function within the DATA steps.

INTRODUCTION

Patient profiles are part of an electronic CTD submission to regulatory agencies and can also be used to reconcile a study database with the CRF as part of a database audit. The more the individual patient listings are similar to the CRF in format and appearance, the more convenient it is for the data management department to perform data reconciliation tasks. One way to achieve patient profiles is to write a series of PROC REPORT statements, one for each data set, then create a macro loop repeating the PROC REPORT statements for each subject. However, the programming and validation of these programs can be tedious since many data sets and variable names are involved. For example, if there is a typo in a variable name within the COLUMN statement using PROC REPORT, assuming the incorrect name is neither in the data set nor in the compute block, that variable will then have all missing values in the output table. Not only is it time-consuming to verify each variable name used in this type of program, it is also redundant to write PROC REPORT statements for each table in the database. There are papers describing how to generate patient profiles, however the methods provided require a lot of data manipulation. Therefore, we developed a new approach to make programming and code validation much easier and more efficient. The first step in our approach is to define a master excel file that carries all attributes that will be used in the PROC REPORT statements. We then developed a macro (%MPROFILE) to generate PROC REPORT statements automatically for each data set by using CALL EXECUTE. Patient profiles are created in Rich Text Format by using ODS RTF.

MASTER METADATA EXCEL FILE

The SASHELP view VCOLUMN, which stores attribute information for all variables in every data set for each library name specified, was first used to retrieve variable and data set names and variable attribute for use in profile generation. The data set and variables names were then exported to an excel file and stored in two worksheets called TABLE and VARIABLE, respectively.

The excel worksheets were edited in order to organize the data and present only the information to be used in the patient profiles such as titles and headers. The goal is to organize the tables and columns in the same order as they appear in CRF. In addition, multiple tables can be presented on a single page to make the output more compact.

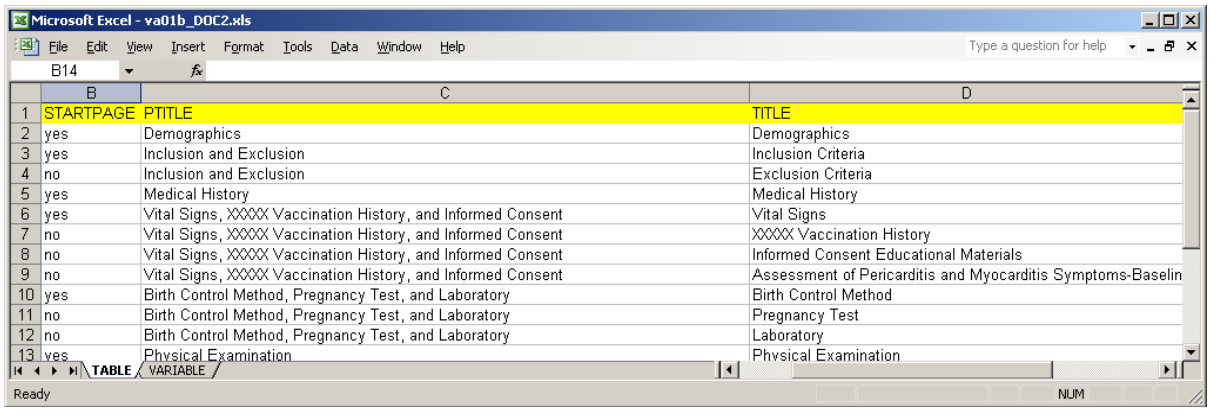
The TABLE worksheet contains four columns: MEMNAME, STARTPAGE, TITLE, and PTITLE. Values in the STARTPAGE column determine if the table should begin a new page or be placed on the same page as the previous table. Values in the PTITLE column define the title used for the entire page while values in the TITLE column define the titles for each individual table. The order of the data set names provided in the MEMNAME column is the order that will be used in the output. An example of the TABLE worksheet is provided in Figure 1.

The VARIABLE worksheet contains the following columns:

MEMNAME	Data set name
TEXTHEADING	Header spanned over several columns
NAME	Variable name
LABEL	Variable label (If the variable label is too long to fit into the available space, the character specified in the SPLIT= option must be used to define how the label

	text should be split.)
COLUMNSEQ	Column sequence within a table
FORMAT	Variable format
ORDEROPTION	Determine the order of the rows of the table. It can take on values: DATA, FORMATTED, FREQ, and INTERNAL. If the value is not specified in the worksheet, then default DATA is used.
CELLWIDTH	Specify the width of the cell in the ODS STYLE= attribute. If the value is missing, the CELLWIDTH attribute will not be specified in PROC REPORT statements.

An example of the VARIABLE worksheet is provided in Figure 2.



Microsoft Excel - va01b_DOC2.xls

File Edit View Insert Format Tools Data Window Help — Type a question for help

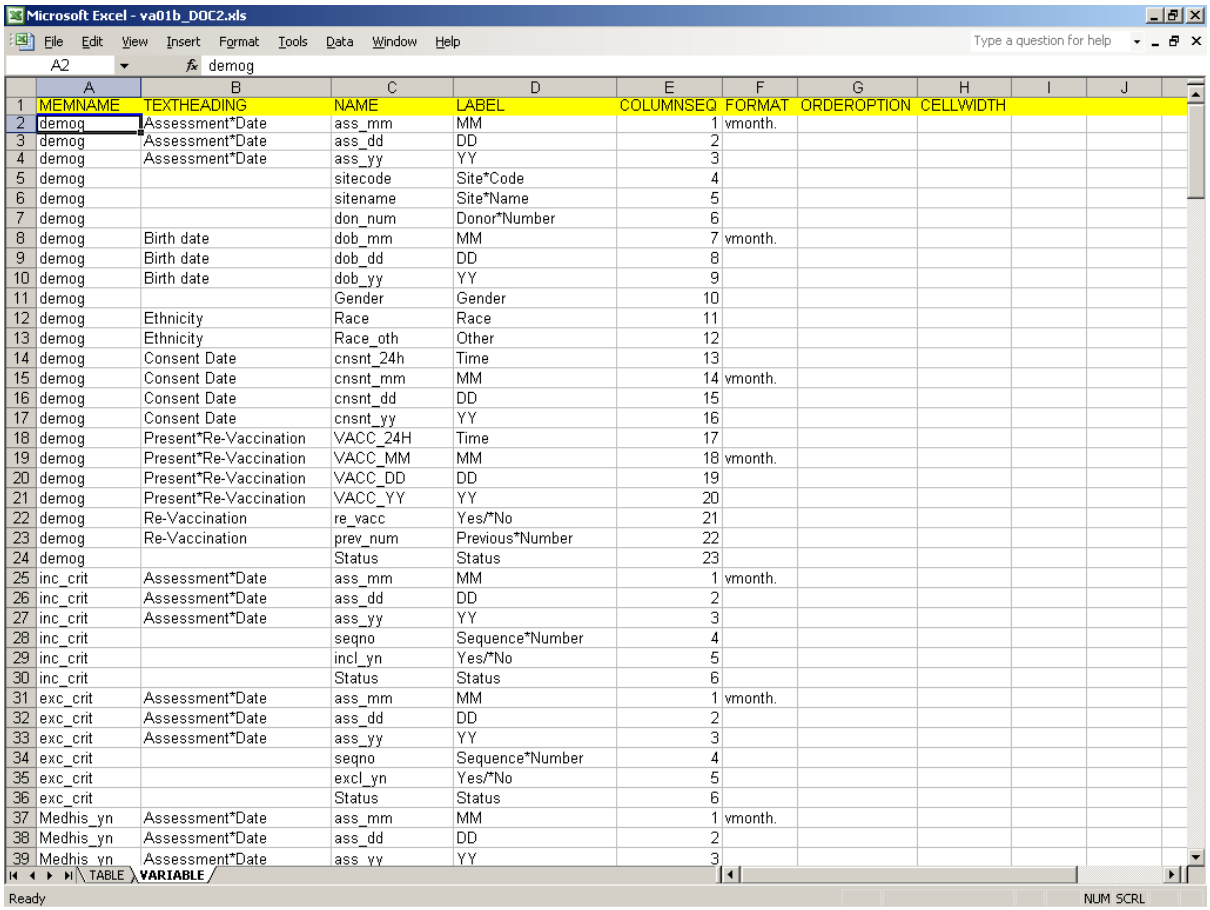
B14

	B	C	D
1	STARTPAGE	PTITLE	TITLE
2	yes	Demographics	Demographics
3	yes	Inclusion and Exclusion	Inclusion Criteria
4	no	Inclusion and Exclusion	Exclusion Criteria
5	yes	Medical History	Medical History
6	yes	Vital Signs, XXXXX Vaccination History, and Informed Consent	Vital Signs
7	no	Vital Signs, XXXXX Vaccination History, and Informed Consent	XXXXX Vaccination History
8	no	Vital Signs, XXXXX Vaccination History, and Informed Consent	Informed Consent Educational Materials
9	no	Vital Signs, XXXXX Vaccination History, and Informed Consent	Assessment of Pericarditis and Myocarditis Symptoms-Baselin
10	yes	Birth Control Method, Pregnancy Test, and Laboratory	Birth Control Method
11	no	Birth Control Method, Pregnancy Test, and Laboratory	Pregnancy Test
12	no	Birth Control Method, Pregnancy Test, and Laboratory	Laboratory
13	yes	Physical Examination	Physical Examination

TABLE VARIABLE

Ready NUM

Figure 1. *An example of the TABLE worksheet*



Microsoft Excel - va01b_DOC2.xls

File Edit View Insert Format Tools Data Window Help — Type a question for help

A2 demog

	A	B	C	D	E	F	G	H	I	J
1	MEMNAME	TEXTHEADING	NAME	LABEL	COLUMNSEQ	FORMAT	ORDEROPTION	CELLWIDTH		
2	demog	Assessment*Date	ass_mm	MM	1	vmonth.				
3	demog	Assessment*Date	ass_dd	DD	2					
4	demog	Assessment*Date	ass_yy	YY	3					
5	demog		sitecode	Site*Code	4					
6	demog		sitename	Site*Name	5					
7	demog		don_num	Donor*Number	6					
8	demog	Birth date	dob_mm	MM	7	vmonth.				
9	demog	Birth date	dob_dd	DD	8					
10	demog	Birth date	dob_yy	YY	9					
11	demog		Gender	Gender	10					
12	demog	Ethnicity	Race	Race	11					
13	demog	Ethnicity	Race_oth	Other	12					
14	demog	Consent Date	cnsnt_24h	Time	13					
15	demog	Consent Date	cnsnt_mm	MM	14	vmonth.				
16	demog	Consent Date	cnsnt_dd	DD	15					
17	demog	Consent Date	cnsnt_yy	YY	16					
18	demog	Present*Re-Vaccination	VACC_24H	Time	17					
19	demog	Present*Re-Vaccination	VACC_MM	MM	18	vmonth.				
20	demog	Present*Re-Vaccination	VACC_DD	DD	19					
21	demog	Present*Re-Vaccination	VACC_YY	YY	20					
22	demog	Re-Vaccination	re_vacc	Yes/*No	21					
23	demog	Re-Vaccination	prev_num	Previous*Number	22					
24	demog		Status	Status	23					
25	inc_crit	Assessment*Date	ass_mm	MM	1	vmonth.				
26	inc_crit	Assessment*Date	ass_dd	DD	2					
27	inc_crit	Assessment*Date	ass_yy	YY	3					
28	inc_crit		seqno	Sequence*Number	4					
29	inc_crit		incl_yn	Yes/*No	5					
30	inc_crit		Status	Status	6					
31	exc_crit	Assessment*Date	ass_mm	MM	1	vmonth.				
32	exc_crit	Assessment*Date	ass_dd	DD	2					
33	exc_crit	Assessment*Date	ass_yy	YY	3					
34	exc_crit		seqno	Sequence*Number	4					
35	exc_crit		excl_yn	Yes/*No	5					
36	exc_crit		Status	Status	6					
37	Medhis_yn	Assessment*Date	ass_mm	MM	1	vmonth.				
38	Medhis_yn	Assessment*Date	ass_dd	DD	2					
39	Medhis_yn	Assessment*Date	ass_yy	YY	3					

TABLE VARIABLE

Ready NUM SCRL

Figure 2*. An Example of the VARIABLE worksheet*

%MPROFILE MACRO:
The macro definition is as follows:

%mprofile(doc=, outpath=, dsubj=,vsubj=, split=);

The parameter DOC specifies the path and name of the master metadata excel file. The value of OUTPATH provides the path where the output should be stored. Profiles will be named with subject ID and stored in that path. DSUBJ specifies the data set name that contains the subject ID, and VSUBJ specifies the variable name used for the subject ID. SPLIT specifies the character that will be used in the SPLIT= option of the PROC REPORT statement.

The macro first imports the metadata excel file then merges the TABLE and VARIABLE worksheets into one SAS data set called _DOC. There are two %DO loops in the macro. The first %do loop chooses one subject in each run. The second %DO loop is nested within the first loop and chooses one table for each run. If there is no data in that table for the specified subject, the output will display 'No data in this table'. Otherwise, CALL EXECUTE routines are used to write the PROC REPORT statements automatically using the information provided in the macro variables and _DOC dataset. An example of the output generated is provided in Figure 3 and Figure 4.

Demographics

Subject #: XX-XXX-846

Table 1 : Demographics

Assessment Date						Birth date				Ethnicity		Consent Date				Present Re-Vaccination				Re-Vaccination		
MM	DD	YY	Site Code	Site Name	Donor Number	MM	DD	YY	Gender	Race	Other	Time	MM	DD	YY	Time	MM	DD	YY	Yes/ No	Previous Number	Status
Jul	25	2007	C01	XXXXXX	1234	Jan	12	1977	M	CAUC		09:43	Aug	7	2007	09:55	Aug	7	2007	N/A		0

***Figure 3**. Sample Output*

Birth Control Method, Pregnancy Test, Laboratory

Subject #: XX-XXX-846

Table 10 : Birth Control Method

No data in this table

Table 11 : Pregnancy Test

No data in this table

Table 12 : Laboratory

Assessment Date				Collection Date										
MM	DD	YY	Test	MM	DD	YY	Greater or Less	Result	Units	Text Values	Within Normal Range?	CS?	Report Attached	Status
Jul	25	2007	CD4+Cell Count	Jul	25	2007		569	UL		Y		Y	0
Jul	25	2007	CD4%	Jul	25	2007		37.9	%		Y		Y	0
Jul	25	2007	IgA	Jul	25	2007		122	MG/DL		Y		Y	0

***Figure 4**. Sample Output for empty data sets*

The advantage of this program is that it writes PROC REPORT statements by itself. The Programmer needs only spend a little time editing the metadata excel file to specify the output format. Once the macro program has been validated for use, the Excel worksheets are all that require verification. Excel provides the flexibility for the Programmer to remove or add tables and variables as well as organize the output with very little effort and without compromising the validated state of the SAS program.

CONCLUSION

The %MPROFILE macro is an easy and efficient way to generate patient profiles that follow the order in which the data is collected on the CRF. Using Excel to define data set names, variable names and their corresponding labels and formats allows the Programmer the flexibility to make changes to the format of the patient profiles without having to edit the SAS program itself. In addition, the macro uses the CALL EXECUTE routine to automatically generate PROC REPORT statements, thus eliminating the need for repetitive coding.

CONTACT INFORMATION

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

Suwen Li
Cangene Corporation
155 Innovation Drive
Winnipeg, Manitoba
Canada R3T 3K1
(204) 275-4152
sli@cangene.com

APPENDIX

```
%macro mprofile(doc=, outpath=, dsubj=,vsubj=, split=);
proc import out= work._table
   datafile= "&doc..xls"
   dbms=excel2000 replace;
   sheet="table";
   getnames=yes;
run;

proc import out= work._variable
   datafile= "&doc..xls"
   dbms=excel2000 replace;
   sheet="variable";
   getnames=yes;
run;

/*assign sequence number to each table and combine table's sequence number into
table title*/
data _table;
   length title1 $100 memname $32;
   set _table(rename=(title=title1));
   where memname^=' ' ;
   memname=upcase(memname);
   tableseq=_n_;
   title=compbl('Table '||left(put(_n_,8.))||': '||title1);
   drop title1;
run;

/*Set default option of ORDER= to DATA */
data _variable;
   length orderoption $10 memname $32;
   set _variable;
   where memname^=' ' and name^=' ';
   memname=upcase(memname);
   if orderoption=' ' Then orderoption='DATA';
run;

proc sort data=_variable;
   by memname;
run;

proc sort data=_table;
   by memname;
run;

data _doc;
   merge _table(in=a) _variable (in=b);
   by memname;
run;

proc sort data=_doc out=_DOC1;
   by tableseq columnseq;
run;

/*Assign data set names to a set of macro variables*/
data _NULL_;
   set _doc1 end=last;
```

```
   call symput('namdatasets'||trim(left(put(tableseq,best.))),     memname);
   if last then call symput('numdataset', tableseq);
run;

%let numdataset=&numdataset;

*****************************************************************;
*Below program use CALL EXECUTE to wrote PROC REPORT for each table;
*****************************************************************;
proc sql noprint;
   select count(*) into: totsubj
   from &dsubj;
   %let totsubj=&totsubj;
   select &vsubj into: subj1 -:subj&totsubj
   from &dsubj;
quit;

*----------------------------------------------------------------;
*Begin to create each patient profile;
*----------------------------------------------------------------;
%do j=1 %to &totsubj;
%let outname=&&subj&j;
data _null_;
   call execute ('ods rtf file="&outpath.\&outname..rtf" ;');
run;

%do i=1 %to &numdataset;
%let datset=&&namdatasets&i;
/*keep only one table's contents*/
data _doc3;
   set _doc1;
   where upcase(memname)=upcase("&datset");
run;

%let ptitle=;
%let startpage=;
%let ttitle=;
proc sql noprint;
   select distinct ptitle, startpage, title into:ptitle,   :startpage,:ttitle
   from _doc3;
quit;

data _null_;
   call execute ('ods rtf startpage=&startpage;');
run;

title1 j=c "&ptitle";
title2 j=l h=1.0 "Subject #: &&subj&j";

/*generate subdata set only for one subject from the specified  table*/
data _temp;
   set &&namdatasets&i;
   where &vsubj="&&subj&j";
run;

data _null_;
   dsid=open('_temp','i');
   cnt=attrn(dsid,'nobs');
   call symput('cnt',put(cnt,8.));
   rc=close(dsid);
run;
```

```
   /*if the dataset is empty, write NO DATA IN THIS TABLE in that table*/
   %if &cnt = 0 %then %do;
      data no;
         desc='No data in this table';
      run;

      data _null_;
          call execute('proc report data=no nowd
style(header)={background=white};');
          call execute('column ("&ttitle" desc);');
          call execute('define desc/ " " width=36 left;');
          call execute('run;');
          call execute ('footnote1 j=r
          "{\field{\*\fldinst{\b\i PAGE}}}\~{\b\i of}\~{\field{\*\fldinst{\b\i
   NUMPAGES}}}";');
      run;
   %end;

   /*if the dataset is not empty then generate table*/
   %else  %do;
      data _doc4;
         length hold $1000 defname $1000 format $32;
         set _doc3;
         retain hold ' ';
         if _N_=1 then do;
         flag=1;
         hold=textheading;
         end;
         if hold ^= textheading then do;
            flag+1;
            hold=textheading;
         end;

         if format ne ' ' then do;
            defname=compbl('define ' ||trim(left(name))||'/display flow '
            ||' '||'format='||trim(left(format))
            ||' '||'order='||trim(left(orderoption)));
         end;
         else do;
         defname=compbl('define ' ||trim(left(name))||'/display flow '
                   ||' '||'order='||trim(left(orderoption)));
         end;
      run;

      proc sort data=_doc4;
         by flag;
      run;

   /*Use data steps to automatically write PROC REPORT*/
      data _null_;
         set _doc4 end=last;
         by flag;
         length vc $1000 define $1000  vd $10000;
         retain vd  ' ' vc  ' ';
         if cellwidth =' ' then do;
            define=trim(left(defname))||' left"' ||trim(left(label))||'" width=20; ';
         end;
         else if cellwidth ^=' ' then do;
            define=trim(left(defname))||' left"' ||trim(left(label))||
                              '" width=20
```

```
style(column)=[cellwidth='||cellwidth||'];';
      end;

      if _n_=1 then do;
         call execute ('proc report data=work._temp nowd center split="&split"
       style(header)={background=white};');
         call execute ('column( ');
         vd=define;
      end;
      else do;
         vd=trim(left(vd))||' '||trim(left(define));
      end;

      if first.flag then vc=name;
      else vc=trim(left(vc))||' '||trim(left(name));

      if last.flag then do;
         if textheading ne ' ' then call execute ('("'||trim(left(textheading))||'"
         '||vc||')');
         else call execute (vc);
      end;

      if last then do;
         call execute (');');
         call execute(vd);
         call execute('compute before _page_ / style=[font_weight=bold
font_size=12pt];');
         call execute('line "'||trim(left(title))||'";');
         call execute('endcomp;');
         call execute ('run;');
         call execute ('footnote1 j=r
     "{\field{\*\fldinst{\b\i PAGE}}}\~{\b\i of}\~{\field{\*\fldinst{\b\i
NUMPAGES}}}";');
      end;
   run;
%end;
%end;
%end;

data _null_;
   call execute('ods rtf close;');
run;

%mend mprofile;
```