PharmaSUG 2015 - Paper BB03

A Toolkit to Create a Dynamic Excel Format Metadata to Assist SDTM Mapping Process

Huei-Ling Chen, HLC Analytics Inc., Edison, NJ
Helen Wang, Sanofi, Bridgewater, NJ

ABSTRACT

SDTM (Study Data Tabulation Model) dataset structure is a recommended standard data format when sponsors (e.g. pharmaceutical companies) submit a new drug application to the FDA. Data management teams and clinical SAS programming teams frequently encounter data mapping tasks to convert raw datasets to SDTM structured datasets. Having a toolkit to quickly summarize the raw datasets into a metadata in Excel spreadsheet format can be helpful to enable the mapping tasks. This paper demonstrates a toolkit which automatically searches the entire designated SAS data library and creates a dynamic Excel format metadata to summarize the data. This metadata has two main features: 1) one overview spreadsheet to list out all datasets embedded with a hyperlink so it can direct to individual dataset spreadsheet; 2) for each dataset, there is an individual spreadsheet presenting the variables attributes. In addition, there is a link directs back to the overview spreadsheet.

KEYWORDS

ODS TAGSETS EXCELXP, SASHELP LIBRARY, PROC CONTENTS, METADATA, SDTM MAPPING

INTRODUCTION

Utilizing knowledge from ODS TAGSETS EXCELXP, SASHELP LIBRARY, and PROC CONTENTS, this paper presents some tips when creating a dynamic excel formatted metadata workbook. One useful application of such a metadata workbook is to assist the project manager of clinical study programming team to have a clear overview on SDTM mapping process.

The excel workbook has one overview spreadsheet to list out all datasets in the designated SAS data library, and each dataset in the library has its own spreadsheet. In addition, in the overview spreadsheet, each cell is embedded with a hyperlink direct to individual dataset spreadsheet in the individual spreadsheet. The header is also equipped with a hyperlink direct to the overview spreadsheet.

KEY SYNTAX 1: RETRIEVE ALL DATASET / VARIABLE NAME FROM SASHELP.VCOLUMN

SAS provides dictionary tables as metadata for the active datasets. SASHELP.VCOLUMN is an associated PROC SQL view to the SAS provided Dictionary table. SASHELP.VCOLUMN contains information about variables in all known datasets. The following code uses VCOLUMN to retrieve the LIBNAME="&INLIB" and MEMTYPE="DATA" records. Two temporary datasets, **META** and **A**, are derived and later serves as a reference source for the individual dataset summary and overview dataset summary, respectively.

```
data META;
     set sashelp.vcolumn(where=(libname="&inlib" and memtype="DATA" ) );
run;

proc freq data=META noprint;
     table  memname / out=A;
run;
```

Utilizing the temporary dataset **A**, count number of datasets in the designated library. Assign each dataset with an individually corresponding macro parameter. This technique can automatically search on each dataset without the need to specify individual file names.

```
proc sql noprint;
     select count(*) into :tot
     from A
     ;
```

```
quit;

proc sql noprint;
     select memname
     into :dat1 - :dat&tot
     from A
     ;
quit;
```

KEY SYNTAX 2: USE PROC CONTENTS TO SUMMARIZE EACH INDIVIDUAL DATASET

```
proc contents data=&inlib..&&dat&i out=meta1 noprint;
run;
```

PROC CONTENTS is a procedure which describes the structure of the data set rather than the data values. OUT statement in the procedure creates a data set where each observation is a variable from the original data set. Selected information from PROC CONTENTS output is presented here: NAME, TYPE, LABEL, FORMAT, FORMATL, INFORMAT, and INFORML.

In addition, this paper uses a data step to check the count of the missing value for each variable. Dataset **META** is served as a reference source for the individual dataset summary here. When the count of missing values for a particular variable equals the number of records in the dataset, this variable is an empty variable. A flag variable BLANKVAR is added to the summary result.

```
< see complete code in Appendix >

* count number of observation;
proc sql noprint;
     select count(*) into :total
     from &inlib..&&dat&i
     ;
quit;

* &char1: all chatogorical variable list - original var name;
* &char2: all chatogorical variable list - new var name ;
proc sql noprint;
     select distinct name
     into :char1 separated by ' '
     from META
     where upcase(type)='CHAR' and memname="&&dat&i"
     ;
     select distinct name
     into :char2 separated by '_1 '
     from META
     where upcase(type)='CHAR' and memname="&&dat&i"
     ;
quit;

* when &char1 is missing (' '), &char2=1;
data final;
     set &inlib..&&dat&i;
     array char1{*} &char1;
     array char2{*} &char2;
     do i=1 to dim(char1);
          if char1{i} = ' ' then char2{i}=1;
     end;
run;

* count number of missing value in a variable;
proc summary data=final;
     var &char2;
     output out=result
```

```
             n = &char1 ;
run;

* add flag variable BlankVar  ;
* when number of observations (&total) =
       number of missing values in a variable, that variable is empty var;
data meta1;
     merge meta1(in=in1)
           result2;
     by memname name;
     if in1;
     if col1=&total and col1 ne 0 then BlankVar='Y';
run;
```

To link to a particular cell in the same workbook, a general format of the link should be followed.

```
#'worksheet-name'!cell-reference
```

```
title link="#'Overview'!A1" "Back to Overview Page";
```

Above code adds a hyperlink in the title of the worksheet to cell A1 of the Overview spreadsheet.

KEY SYNTAX 3: PREPARE AN OVERVIEW WORKSHEET

The overview worksheet would list out dataset name and number of observations. In addition, a flag variable BLANKDAT identifies dataset empty or not.

```
data A;
     length frmshort $20.;
     set A;
     if memname="&&dat&i" then do;
         frmshort="&dcmname";
         NumObs=&total;
         BlankDat='';
         %if &total=0 %then %do;
                BlankDat='Y';
         %end;
     end;
run;
```

A hyperlink in the cell to each dataset worksheet is set up here. Clicking on a dataset name will display the worksheet containing PROC CONTENTS output from KEY SYNTAX 3. For example, clicking on dataset 'AE_M' displays the worksheet "AE_M" (see Figure 2 below).

The general Syntax for hyperlink in the Microsoft Excel:

```
HYPERLINK(link_location,friendly_name)
```

Below code generates the hyperlink to 'xls' extension file name.

```
data b;
     set A;
     tempname="[ListDataContents_&inlib..xls]";
     dataset= '=HYPERLINK("' || trim(left(tempname)) || trim(left(memname))
              || '!A1", "' || trim(left(memname)) ||'")';
run;
```

The value of variable DATASET is like below.

DATASET
=HYPERLINK("[ListDataContents_RDBS.xls]AEI_M!A1", "AEI_M"
=HYPERLINK("[ListDataContents_RDBS.xls]AEJ_M!A1", "AEJ_M"
=HYPERLINK("[ListDataContents_RDBS.xls]AE_M!A1", "AE_M"
=HYPERLINK("[ListDataContents_RDBS.xls]AQL_M!A1", "AQL_M"
…

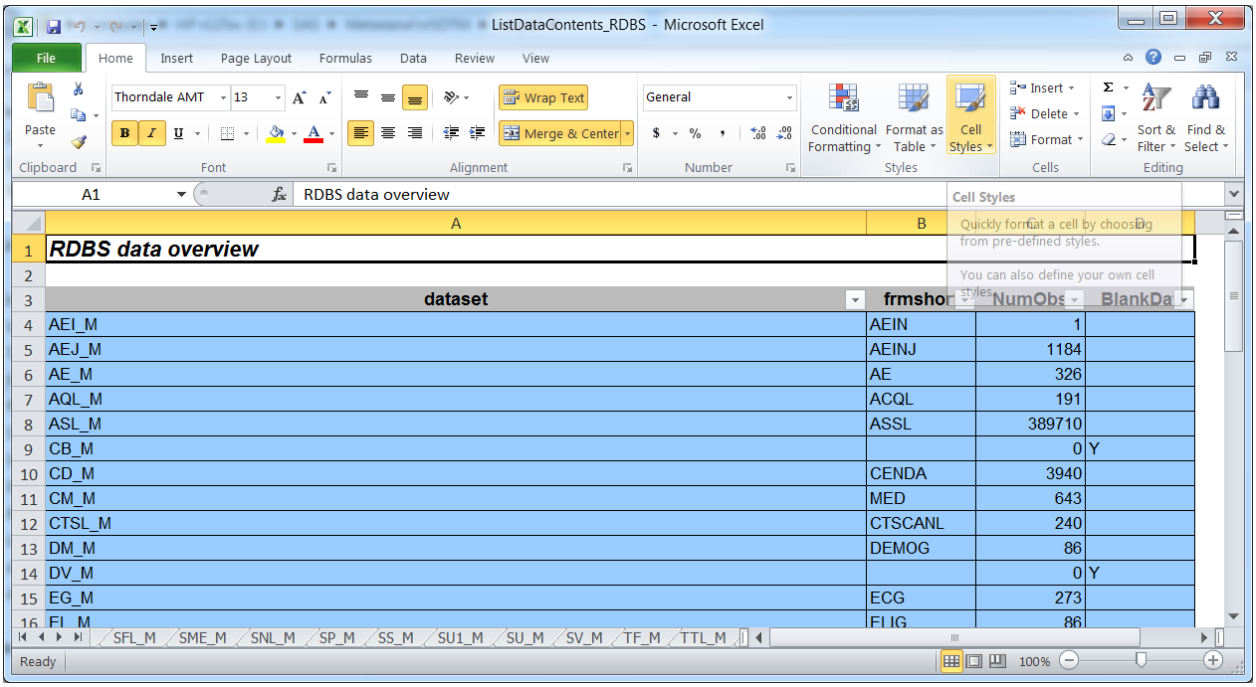
KEY SYNTAX 4: USE ODS TAGSETS EXCELXP TO CREATE A DYNAMIC EXCEL WORKSHEET

With the hyperlink features covered in Key Syntax 2 and 3, a dynamic Excel worksheet is almost ready except that the most important task – initial ODS tagsets.ExcelXP.

```
** initial ods tagsets.ExcelXP ;
ods _all_ close;
ods tagsets.ExcelXP path="&path"
                    file="ListDataContents_&inlib..xml"
                    style=Printer;

** create individual dataset worksheet;
title link="#'&inlib'!A1" "Back to &inlib Main Page";

ods tagsets.ExcelXP options(embedded_titles='yes'
                            embedded_footnotes='yes'
                            print_header=''
                            print_footer=''
                            autofilter='all');

ods tagsets.ExcelXP options(sheet_name="&&dat&i");

< proc print data = meta1; >

** create overview summary worksheet;
ods tagsets.ExcelXP options(sheet_name="&inlib");

< proc print data = b; >

< details on ODS Styles are in the complete code in Appendix >

** close the ods tagsets.ExcelXP;
ods tagsets.ExcelXP close;
```

OUTPUT

The macro would create an excel workbook. In the workbook, there is one overview worksheet and each dataset in the library has its own spreadsheet. Figure 1 below is the overview worksheet. Figure 2 is an example worksheet for the individual dataset.

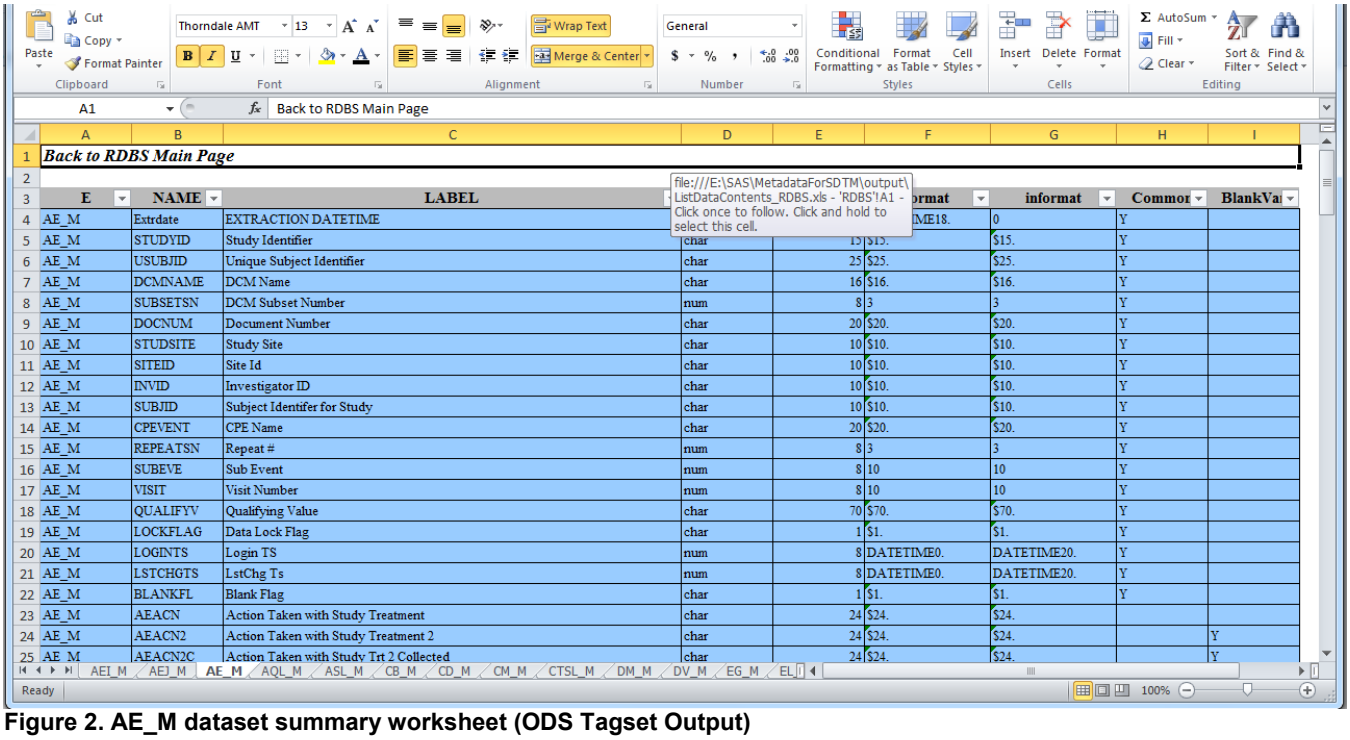


RDBS data overview

dataset	frmshor	NumObs	BlankDa
AEI_M	AEIN	1	
AEJ_M	AEINJ	1184	
AE_M	AE	326	
AQL_M	ACQL	191	
ASL_M	ASSL	389710	
CB_M		0	Y
CD_M	CENDA	3940	
CM_M	MED	643	
CTSL_M	CTSCANL	240	
DM_M	DEMOG	86	
DV_M		0	Y
EG_M	ECG	273	
EL_M	ELIG	86	

Figure 1. RDBS overview worksheet (ODS Tagset Output)

The overview worksheet lists out dataset name (column A), eCRF form short name (column B), number of observations (column C), and a flag variable BLANKDAT identifying dataset empty or not (column D). Each cell in the column A has embedded hyperlinks to direct user to each dataset spreadsheet. It saves the hustle and bustle of scrolling the tab bar left or right to search for the dataset. For example, clicking on dataset 'AE_M' would directly open the worksheet "AE_M"



Back to RDBS Main Page

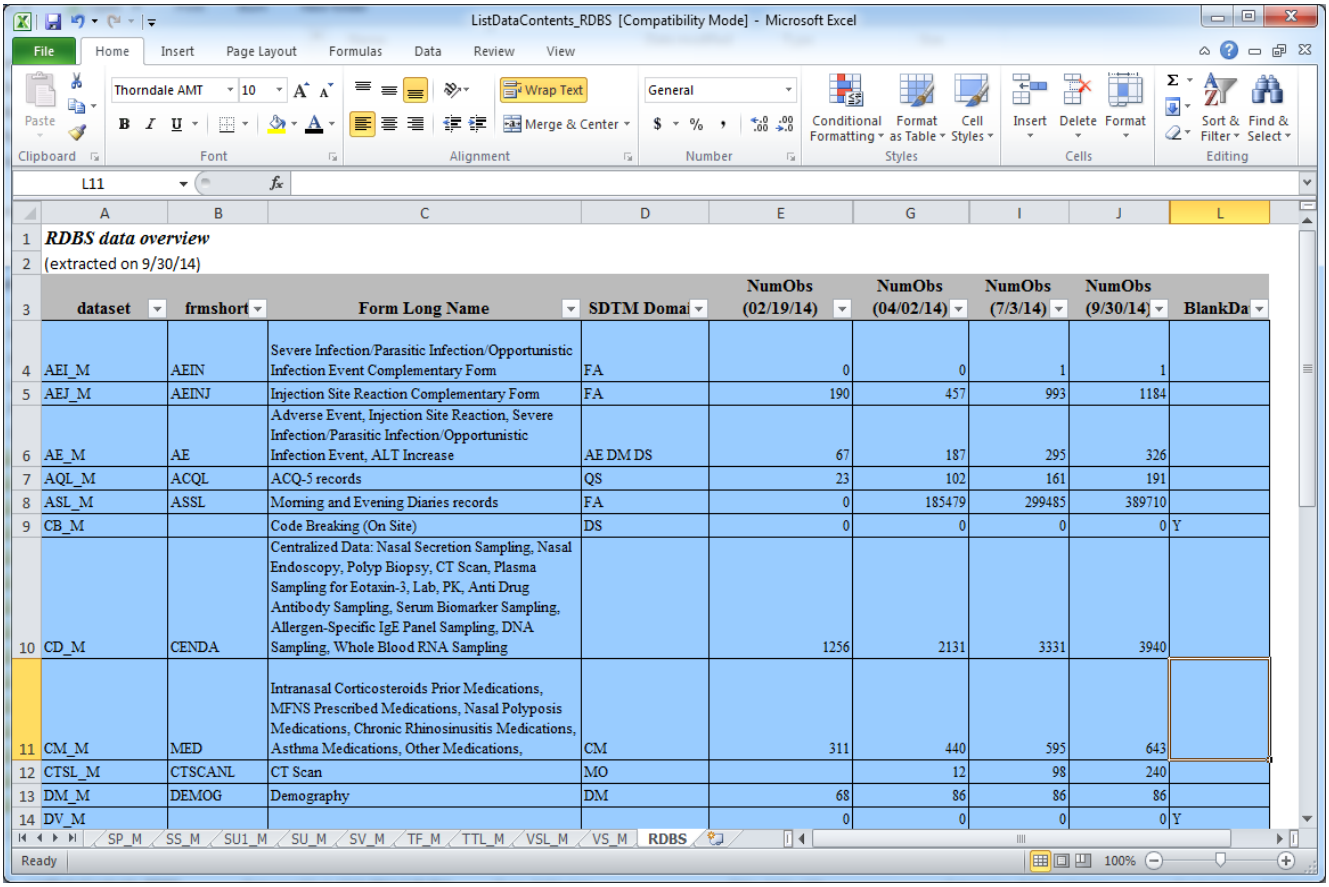
E	NAME	LABEL			informat	Common	BlankVa
AE_M	Extrdate	EXTRACTION DATETIME			0	Y	
AE_M	STUDYID	Study Identifier	char	15 $15.	$15.	Y	
AE_M	USUBJID	Unique Subject Identifier	char	25 $25.	$25.	Y	
AE_M	DCMNAME	DCM Name	char	16 $16.	$16.	Y	
AE_M	SUBSETSN	DCM Subset Number	num	8 3	3	Y	
AE_M	DOCNUM	Document Number	char	20 $20.	$20.	Y	
AE_M	STUDSITE	Study Site	char	10 $10.	$10.	Y	
AE_M	SITEID	Site Id	char	10 $10.	$10.	Y	
AE_M	INVID	Investigator ID	char	10 $10.	$10.	Y	
AE_M	SUBJID	Subject Identifer for Study	char	10 $10.	$10.	Y	
AE_M	CPEVENT	CPE Name	char	20 $20.	$20.	Y	
AE_M	REPEATSN	Repeat #	num	8 3	3	Y	
AE_M	SUBEVE	Sub Event	num	8 10	10	Y	
AE_M	VISIT	Visit Number	num	8 10	10	Y	
AE_M	QUALIFYV	Qualifying Value	char	70 $70.	$70.	Y	
AE_M	LOCKFLAG	Data Lock Flag	char	1 $1.	$1.	Y	
AE_M	LOGINTS	Login TS	num	8 DATETIME0.	DATETIME20.	Y	
AE_M	LSTCHGTS	LstChg Ts	num	8 DATETIME0.	DATETIME20.	Y	
AE_M	BLANKFL	Blank Flag	char	1 $1.	$1.	Y	
AE_M	AEACN	Action Taken with Study Treatment	char	24 $24.	$24.		
AE_M	AEACN2	Action Taken with Study Treatment 2	char	24 $24.	$24.		Y
AE_M	AEACN2C	Action Taken with Study Trt 2 Collected	char	24 $24.	$24.		Y

Figure 2. AE_M dataset summary worksheet (ODS Tagset Output)

Each dataset has its own spreadsheet. Figure 2 is a spreadsheet for dataset AE_M. The worksheet lists out dataset name (column A), variable name (column B), variable label (column C), variable type character or numeric (column D), variable length (column E), variable format (column F), variable informat (column G), a study specific variable COMMON (column H),

and a flag variable BLANKVAR identifying variable with null value or not (column I). The top row of the spreadsheet has a hyperlink which can take user back to the overview worksheet.

APPLICATION



	dataset	frmshort	Form Long Name	SDTM Domai	NumObs (02/19/14)	NumObs (04/02/14)	NumObs (7/3/14)	NumObs (9/30/14)	BlankDa
4	AEI_M	AEIN	Severe Infection/Parasitic Infection/Opportunistic Infection Event Complementary Form	FA	0	0	1	1	
5	AEJ_M	AEINJ	Injection Site Reaction Complementary Form	FA	190	457	993	1184	
6	AE_M	AE	Adverse Event, Injection Site Reaction, Severe Infection/Parasitic Infection/Opportunistic Infection Event, ALT Increase	AE DM DS	67	187	295	326	
7	AQL_M	ACQL	ACQ-5 records	QS	23	102	161	191	
8	ASL_M	ASSL	Morning and Evening Diaries records	FA	0	185479	299485	389710	
9	CB_M		Code Breaking (On Site)	DS	0	0	0	0	Y
10	CD_M	CENDA	Centralized Data: Nasal Secretion Sampling, Nasal Endoscopy, Polyp Biopsy, CT Scan, Plasma Sampling for Eotaxin-3, Lab, PK, Anti Drug Antibody Sampling, Serum Biomarker Sampling, Allergen-Specific IgE Panel Sampling, DNA Sampling, Whole Blood RNA Sampling		1256	2131	3331	3940	
11	CM_M	MED	Intranasal Corticosteroids Prior Medications, MFNS Prescribed Medications, Nasal Polyposis Medications, Chronic Rhinosinusitis Medications, Asthma Medications, Other Medications,	CM	311	440	595	643	
12	CTSL_M	CTSCANL	CT Scan	MO		12	98	240	
13	DM_M	DEMOG	Demography	DM	68	86	86	86	
14	DV_M				0	0	0	0	Y

Figure 3. Modified RDBS overview worksheet

One useful application of such a metadata workbook is to add some additional information and assist the project manager of clinical study programming team to have a clear overview on SDTM mapping process.

In our study, around 50+ raw datasets are to be mapped to 20+ SDTM domains. Multiple eCRF forms could be stored into one dataset. Hence it is important to list out all eCRF form names. The eCRF complete form name is added manually in column C in figure 3. This column can help programmer to associate the data variables and values to the eCRF form.

Some eCRF form would go to one same SDTM domain. Some eCRF form would be mapped to different domains. Adding a column 'SDTM domain' to list out the dataset SDTM destination could present the flow clearly. The SDTM domain is added manually in column D in figure 3.

Data size increases for each extraction in an ongoing study. Keeping track of number of observations can help programmer to trace the progress of the study. In figure 3, 'NumObs' columns are manually kept from previous run worksheet and inserted into the most recent updated sheet. See Column E to J.

CONCLUSION

The work of SDTM mapping can be complicated and overwhelming. When working on SDTM mapping, it is essential to have knowledge on the raw data, e.g. how the raw data structure look like and what the variable meaning is. To enhance knowledge on the raw data, a direct approach is to read eCRF form and to associate the eCRF form with the data variables and values. This paper presents a portable macro which would automatically search the entire designated SAS data library and creates a dynamic Excel format metadata to summarize the data quickly. One of application of this metadata worksheet is to assist a clinical study programming team when working on SDTM mapping projects.

The macro is portable and general and not limited to be used in the clinical study trials. The complete code is listed in the Appendix session.

REFERENCES

- DelGobbo, V. (2007) "Creating Multi-Sheet Excel Workbooks the Easy Way with SAS®," Proceedings of the NESUG 2007 Conference.
- Chen, H. (2010) "ADaM Dataset Checking Toolkit," Proceedings of the PharmaSUG 2010 Conference.

ACKNOWLEDGMENTS

The authors would like to thank the B&P management team of Sanofi-Aventis for their advice on this paper/presentation.

CONTACT INFORMATION

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

Huei-Ling Chen
HLC Analytics Inc.
Edison, NJ 08820
Phone: 908-208-6911
e-mail: huchentw@yahoo.com

Helen Wang
Sanofi
55C-330A, 55 Corporate Drive
Bridgewater, NJ 08807
Phone: 908-981-4752
e-mail: helen.wang@sanofi.com

TRADEMARK

APPENDIX

Example code

```
%macro ListDataContents(path=, inlib = );

    %let inlib=%upcase(&inlib);

    ** retrieve all data/var name from sashelp.vcolumn;
    data meta;
        set sashelp.vcolumn(where=(libname="&inlib" and memtype="DATA" ) );
        run;

    ** count number of datasets in the designated library;
    proc freq data=meta noprint;
        table  memname / out=a;
        run;

    proc sql noprint;
```

```
        select count(*) into :tot
        from a
        ;
        quit;

    %let tot=&tot;

    ** assign each dataset with individually corresponding macro parameters ;
    proc sql noprint;
        select memname
        into :dat1 - :dat&tot
        from a
        ;

    ** initial ods tagsets.ExcelXP ;
    ods _all_ close;
    ods tagsets.ExcelXP path="&path"
                        file="ListDataContents_&inlib..xml"
                        style=Printer;

    ** start summarize dataset and put the contents result into excel spreadsheet, every
dataset has its own spreadsheet;
    %do i=1 %to &tot ;
        %let char1= ;
        %let char2= ;
        %let num1= ;
        %let num2= ;
        %let dcmname= ;

         * count number of observation;
         * &char1: all chatogorical variable list - original var name;
         * &num1: all numerical variable list - original var name;
         * &char2: all chatogorical variable list - new var name ;
         * &num2: all numerical variable list - new var name;
         proc sql noprint;
              select count(*) into :total
              from &inlib..&&dat&i
              ;
              select distinct name
              into :char1 separated by ' '
              from meta
              where upcase(type)='CHAR' and memname="&&dat&i"
              ;
              select distinct name
              into :char2 separated by '_1 '
              from meta
              where upcase(type)='CHAR' and memname="&&dat&i"
              ;
              select distinct name
              into :num1 separated by ' '
              from meta
              where upcase(type)='NUM' and memname="&&dat&i"
              ;
              select distinct name
              into :num2 separated by '_1 '
              from meta
              where upcase(type)='NUM' and memname="&&dat&i"
              ;
              quit;

         %if %length(&char1)=0 %then %let char1=dumc;
         %if %length(&num1)=0 %then %let num1=dumn;
         %let char2 = &char2._1;
         %let num2 = &num2._1;

         * when &char1 is missing (' '), &char2=1;
```

```
* when &num1 is missing (.), &num2=1;
data final;
     set &inlib..&&dat&i;
     dumc='';
     dumn=.;
     array char1{*} &char1;
     array char2{*} &char2;
     array num1{*} &num1;
     array num2{*} &num2;
     do i=1 to dim(char1);
        if char1{i} = ' ' then char2{i}=1;
     end;
     do i=1 to dim(num1);
        if num1{i}=. then num2{i}=1;
     end;
     run;

* count number of missing value in a variable;
proc summary data=final;
     var &char2 &num2;
     output out=result
            n = &char1 &num1;
     run;

proc transpose data=result out=result1;
     run;

data result2;
     set result1;
     if _name_ in ("_TYPE_" "_FREQ_") then delete;
     memname="&&dat&i";
     rename _name_=name;
     run;

proc sort data=result2;
     by memname name;
     run;

* take the dataset content information from PROC CONTENTS;
proc contents data=&inlib..&&dat&i out=meta1 noprint;
     run;

proc sort data=meta1;
     by memname name;
     run;

* add flag variable BlankVar  ;
* when number of observations (&total) = number of missing values in a variable,
  that variable is all blank;
data meta1;
     merge meta1(in=in1 rename=(type=typen format=format_ informat=informa_) )
           result2;
     by memname name;
     if in1;

     if col1=&total and col1 ne 0 then BlankVar='Y';
     if typen=2 then type='char';
     else if typen=1 then type='num';

     format=trim(left(format_)) || trim(left(put(formatl,8.0))) || '.';
     informat=trim(left(informa_)) || trim(left(put(informl,8.0))) || '.';
     run;

* sort the contents of dataset by the original variable order;
proc sort data=meta1;
     by varnum;
```

```
        run;

    * prepare main sheet (&inlib)
    * list out dataset, number of observations, dataset empty or not;
    * when dataset is empty - flag BlankDat  ;
    data a;
        length frmshort $20.;
        set a;
        if memname="&&dat&i" then do;
          frmshort="&dcmname";
          NumObs=&total;
          BlankDat='';
    %if &total=0 %then %do;
          BlankDat='Y';
    %end;
        end;
        run;

    ** export dataset content to excel file ;
    ** add a hyperlink to the &inlib main page;
    title link="#'&inlib'!A1" "Back to &inlib Main Page";
    footnote ' ';

    ods tagsets.ExcelXP options(embedded_titles='yes'
                                embedded_footnotes='yes'
                                print_header=''
                                print_footer=''
                                autofilter='all');

    ods tagsets.ExcelXP options(sheet_name="&&dat&i");

    proc print data=meta1 noobs style(Header)=[just=center];
        var memname name label type length format informat
            blankvar / style(Column)=[background=#99ccff];
        run;
        quit;

%end;

    * update main sheet (&inlib)
    * add a hyperlink for each dataset name to link to each individual dataset sheet;
    * set the hyperlink to 'xls' extension file name;
    * open excel, save the file to xls;
    data b;
       set a;
       tempname="[ListDataContents_&inlib..xls]";
       dataset= '=HYPERLINK("' || trim(left(tempname)) || trim(left(memname)) ||
                '!A1", "' || trim(left(memname)) ||'")';
       run;

    *** print the main page to excel file;
    title "&inlib data overview ";
    ods tagsets.ExcelXP options(embedded_titles='yes'
                                embedded_footnotes='yes'
                                print_header=''
                                print_footer=''
                                autofilter='all');

    ods tagsets.ExcelXP options(sheet_name="&inlib");

    proc print data=b noobs style(Header)=[just=center];
        var dataset
            numobs blankdat / style(Column)=[background=#99ccff];
        run;
        quit;
```

```
        ** close the ods tagsets.ExcelXP;
        ods tagsets.ExcelXP close;

%mend ListDataContents;
```