

Generation and maintenance of the Data specification document

Melodie Bobineau, VALOS, Genoa, Italy
Dmitri Petratchenko, VALOS, Genova, Italy

ABSTRACT

Helpful for a lot of people involved in the study, the data specification document can be hard to keep up-to-date, especially when the SAP is in working progress.

One solution to avoid it is to write the variables comments within the dataset program code. They can then be extracted and outputted in a dataset in order to be merged with the information of the variables, which has been previously extracted by a PROC CONTENTS. In this way the data specification document can be simply and quickly created outputting this "data specification dataset" into a RTF or PDF file document.

This process decreases substantially the time and efforts spent on the creation of the dataset specification documentation. It also makes it easier and faster to update when needed. Not only can this method be used for the documentation of the analysed dataset which is requested but also for any other dataset.

INTRODUCTION

For the programmers who produce the outputs, the CRO handing over a study to their client, a new study similar to a previous one, new programmers/statisticians who need to familiarized with an on-going study, or simply for the authorities, the data specification document can be a very useful document. Even more, that nowadays most of the data manipulation is done during the creation of the dataset when using the CDISC standards; which make the programming of the outputs mostly resulting in relatively simple summary calculations and layouts.

However we need to be sure to select the correct variable(s) and that their transformation or calculations correspond to the current SAP. But when done manually, the documentation might be complicated to update or review before final version; specially, when changes have been done in the SAP or after feedback from the QC programmer. What solution do we have? How could we shorten the time and efforts used to create it? How could we make the process more automatic?

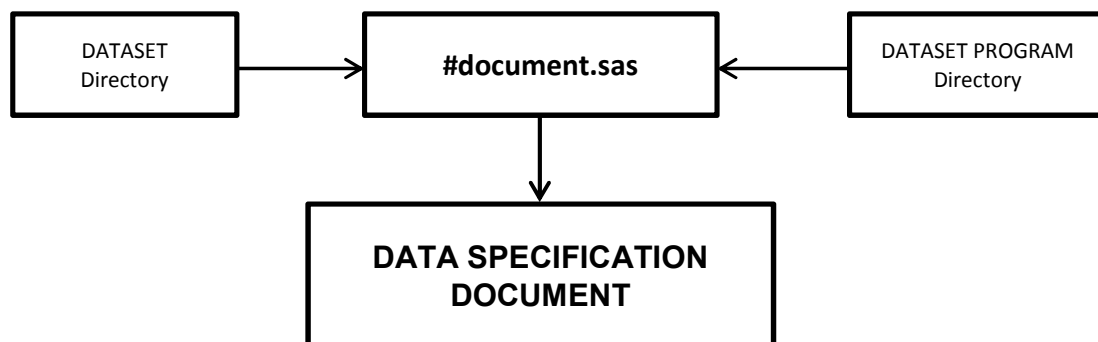
We will see how the production of this document can be easily shorten in time and effort by simply putting standardized comments within the dataset programming code where the variables are created. After having a quick overview of the method used including how to write the variable comments; we will more specifically look at the process used to collect and output in the document all the information about the datasets and theirs variables.

OVERVIEW

The Following chart shows the simplicity of the process. The dataset directory contains the programs used to create the datasets aims to be describe in the data specification document. The programs used to create those dataset are stored within a dataset directory. Both directories can also contain unrelated files to the document.

Most the process is done in the program **#document.sas** which basically read the comments written in the dataset programs and extracts the dataset information.

Finally all details are merged and outputted directly into a document via a **define_rtf** or **define_pdf** program or other as preferred.



PhUSE 2012

WRITING COMMENTS FOR STANDARD AND NON-STANDARD VARIABLES.

The comments and calculations details of standard variables are written within the data-set program code as below. It can be written next to the variable assignment or at the start of the data manipulation needed to create the variable, to the preference the programmer or convention of the company.

```
***@variable-name@@comments@***; or /***@variable-name@@comments@***/
```

Example:

```
***@CQ_SCORE@@ Derived: sum of CQ_Q1-CQ_Q7***;  
CQ_SCORE = sum(of CQ_Q1-CQ_Q7);
```

However putting the comment for series of variables (ex: *num1*, *num2*, *mum3*, ...) could be tricky. Hence a solution has been found to avoid time consuming writing a multiple of time the same kind of comments. The same principal could be easily used for variables created with an array statement.

```
*@@variable-name-base&i@@ comments @@;
```

Example:

```
***@cq_q1n-cq_q7n@@Derived: sum of variables QC_0#iA to QC_0#iD @***;  
%do i=1 %to 7;  
    Array CQ_Q&i{4} QC_0&iA QC_0&iB QC_0&iC QC_0&iD;  
    CQ_Q&i.N = sum(of CQ_Q&i{*});  
%end;
```

PROCESS

The first part of the program **#document.sas** contains all the macro variables and LIBNAMES needed. In this way the program is easily updated when a new document request has been ask for the same study or another. Macro variable **&repconv** is for the path where are the dataset programs, **&ppt** design where the programs for the document creation are, **&final** where the document should be outputted and **&doc** is the document file name. LIBNAME **data** is for the location of the dataset files and **datadoc** where a shot of the dataset should be saved.

#document.sas

```
OPTIONS MISSING=' ' source notes mprint nomacrogen errors=0 compress=yes spool;  
  
%let disk=S;  
%let rep=study_folder;  
%let prot=protocol;  
  
%let repconv=%str(Study_folder\protocol\ADB_extract\pgm\);  
%let ppt =%str(&disk:\&rep\documents\ADB_pgm);  
%let final=%str(&disk:\&rep\documents\ADB_doc);  
%let doc=doc_spec;  
%let tit=%str(Study &prot Analysis Datasets Specifications);  
  
libname data "&disk:\&rep\ADB_data";  
libname datadoc "&disk:\&rep\ADB_doc";
```

PhUSE 2012

DATA SELECTION

If for any reason the datasets need to be reduced, they can be copy in another location prior alteration. For example: selected patients, cutoff date, less variables, etc. It's also possible during this step to protect the dataset of further modifications by protecting them.

#document.sas (simple example)

```
proc datasets library=data;
  copy out=datadoc;
  select ACQ AES AQL DEM;
run;

*< extra code for data selection >;
```

READING THE VARIABLE COMMENTS

Next, the program read the comments previously writing in the program files as suggested above by calling the program **read_conversion.sas**.

#document.sas

```
%include    "&ppt\read_conversion.sas" ;
```

The program start by enumerating the files stored in the dataset program directory (*&repconv*) and output the information (file name, extension) in the file **work.dirfiles** containing the variables: *outfiles* (full program names), *name* (program without extension) and *exten* (file extension). Only files with **.sas** extension are selected excluding the ones not containing for sure any variable comments: see the **if** STATEMENT.

The full and short name are then stored in macro variables (**&namei**, **&sigli**) and the total number of programs is kept in the macro variable **&fin** which was set-up to zero at the start of the program.

read_conversions.sas

```
OPTIONS MISSING=' ' source notes mprint nomacrogen errors=0 compress=yes;

%let fin=0;

filename dircmd pipe "dir /b &repconv."; *** Check DOS options /A:-D /B /O:D ***;

data dirfiles(Keep= outfiles name exten);
  length outfiles $100 name exten $20;
  infile dircmd missover length=length;
  input @;
  input bigline $varying200. length;
  Outfiles = trim(left(bigline));
  name=SCAN(outfiles, 1, ".");
  exten=SCAN(outfiles,2, ".");
  if lowercase(exten)='sas' and lowercase(name)^='s_' and name^='#'
    and lowercase(name) not in ('copydate','libname','formats');
run;

data dirfiles;
  set dirfiles end=eof;
  call symput("name"||compress(_N_),Outfiles);
  call symput("sigl"||compress(_N_),name);
  if eof then do;
    call symput("fin",_N_);
  end;
run;
```

PhUSE 2012

The following code which contains the macro **%read_sas** allows us to have the dataset **work.list** which contains the dataset names, labels, and comments of their variables.

Three kinds of datasets are created during this step: **work.list&i** which contains the full code of each program file (one observation = one line of code). From those files, the **work.temp&i** datasets are created. They contain only the lines/observations which include variable comment. In case of iterative variables the comments are incremented producing the **work.ulist&i** datasets.

Those last datasets are then merged together in order to create the **work.list** dataset, ready to be merged the dataset contents information.

read_conversions.sas

```
%macro read_sas;
  data list;
    length text $300.; text=' ';
    stop;
  run;

  %do i=1 %to &fin;*finish at the end of the macro;
    data list&i;
      infile "&rep.&&name&i" MISSOVER PAD lrecl=32767;
      informat text $300.;
      format text $300. ;
      input @1 text $char300.;
      nn=&i;
      n=_n_;
    run;

*Scanning for comments;
*****;
  data temp&i;
    set list&i;
    length source source2 var $20 comm $300;
    retain source2 flag var;
    if _N_=1 then do; source2=' '; flag=.; var=' '; end;
    if flag=2 then flag=.;
    source=scan("&&name&i",1,'. ');
    if lowercase(source)='ADB_conv' or lowercase(source)='ADB_deriv' then do;
      if left(text)='%macro' and left(reverse(text))=';tsop' then do;
        source2=reverse(scan(substr(left(reverse(text)),6),1,' '));
        if lowercase(source)='ADB_deriv' then
          source2=reverse(scan(substr(left(reverse(text)),7),1,' '));
      end;
    end;
    else source2=source;
    if INDEX(text,'***@') then do;
      flag=1;
      var=scan(substr(text,INDEX(text,'***@')+4),1,'@');*Read the variable name;
      comm=scan(substr(text,INDEX(text,'***@')+4),2,'@');
    end;
    if INDEX(text,'@***') then do;
      t=1;
      if comm=' ' then do; comm=scan(text,1,'@'); t=2; end;
      flag=2;
    end;
    if flag=1 and comm=' ' then comm=text;
    comm=tranwrd(comm,'***',' ');
    comm=tranwrd(comm,'***',' ');
    comm=left(comm);
    if flag^=.;
  run;
```

PhUSE 2012

read_conversions.sas

(continued)

```
*Iterating the comment for incremental variables ;
*****;

data ulist&i ;
  set temp&i;
  length tcomm $ 300;
  if index(var,'-') ne 0 then do;
    varn = compress(var,'-0123456789','k');
    vstartc = left(scan(varn,1,'-'));
    vstart = vstartc*1;
    vstop = scan(varn,2,'-')*1;
    end;
    if vstart ne . and vstop ne . then do;
      tcomm = comm;
      tvar = var;
      numit = (vstop - vstart) + 1;
      do i=1 to numit;
        vnumc = left(put((vstart + i - 1),2.));
        var = tranwrd(scan(tvar,1,'-'),trim(vstartc),trim(vnumc));
        if index(tcomm,'#i') ne 0 then comm=tranwrd(tcomm,'#i',trim(vnumc));
        output;
      end;
    end;
    else output;
  run;

*Merging the comment datasets;
*****;

data list;
  set list ulist&i;
  line=_N_;
  line2=0;
  if lowercase(source)='ADB_conv' then line2=1;
  if lowercase(source)='ADB_deriv' then line2=2;
  var=upcase(var);
  source2=upcase(source2);
  run;
  %end;*end of do loop;
%mend read_sas;

%read_sas;

proc sort data=list; by var line2 line; run;

data list;
  set list;
  by var line2 line;
  length comment $3000;
  retain comment;
  if first.var then comment=' ';
  comment=trim(comment)||" "||left(comm);
  comment=left(comment);
  if last.var then output;
run;
```

PhUSE 2012

MERGING THE COMMENTS WITH THE DATA INFORMATION

The dataset and variables information are extracted via a PROC CONTENTS into a dataset named **work.lblsaf** which in this example contains the dataset names and label, their variable position, name, label, type and length. Other info as for example variable format could also be added.

After some manipulations, the dataset is merged with the previously produced dataset **work.list**. The dataset **work._datadef** (see annex 1 page 8) is created where all the data definition information is ready to be outputted in order to create the data specification document.

#document.sas

```
proc contents data=datadoc._ALL_ noprint
    out=lblsaf(keep= MEMNAME MEMLABEL VARNUM NAME LABEL TYPE LENGTH);
run;

proc sort data=lblsaf;
    by MEMNAME NAME;
run;

data all;
    retain dataset variable VARNUM label type_ length;
    length dataset $255 variable $255 label $255;
    set lblsaf;
    keep dataset description variable label type_ length varnum;
    dataset=upcase(MEMNAME);
    label dataset='DATASET';
    variable=upcase(name);
    label variable= 'VARIABLE';
    label =upcase(label);
    label label='LABEL';
    if type=2 then type_='C';
    if type=1 then type_='N';
    label type_='TYPE';
    label length= 'LENGTH';
    rename MEMLABEL=Description;
run;

proc sort data=all;
    by dataset variable;
run;
proc sort data=list (rename=(source2=dataset var=variable));
    by dataset variable;
run;

data all (keep=dataset dsord variable varnum label type_ length comment)
    err_doc;
    merge all (in=a) list (in=b);
    by dataset variable;
    retain dsord;
    label comment='COMMENTS';
    if first.dataset then dsord=sum(dsord,1);
    if a then output all;
    if b and not a then output err_doc;
run;

data _datadef;
    set all;
    by dataset variable;
run;
```

PhUSE 2012

DOCUMENT CREATION

As seen on annex 1 page 8, the dataset **work_datdef** contains all the information needed to create the data specification document. It can be instantaneously produce using a simple PROC PRINT, PROC REPORT, or more refined using a **define_rtf/define_pdf** program including a PROC TEMPLATE.

```
%include "&ppt\define_rtf.sas" ;  
  
%define_rtf(work,L, "&final.\&doc..rtf", %str(Study &prot Analysis Data Sets));
```

CONCLUSION

Who never lost time creating the data specification document or even more checking and updating it? Trying to see what has been changed and need to be updated? And wish it could be done almost in an instant?

Writing the comment/calculations details within the dataset program itself decreased subsequently the time spent updating the document. It allow us to combined the review of the code and document simultaneously. And hence avoids discrepancy between the two.

From a very simple idea and without practically any additional time and resources, the data specification document can be produced with good quality. If we needed some words to describe it, it would be: "simple but very useful"

CONTACT INFORMATION

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

Melodie Bobineau
VALOS
Via Antonio Gramsci, 1/13
16126 GENOVA
Italy
Tel: +39 010 40771822
Fax: +39 010 40771822
Email: melodie.bobineau@valos.it
Web: <http://www.valos.it>

Dmitri Petratchenko
VALOS
Via Antonio Gramsci, 1/13
16126 GENOVA
Italy
Tel: +39 010 40771822
Fax: +39 010 40771822
Email: Dmitri.petratchenko@valos.it
Web: <http://www.valos.it>

Brand and product names are trademarks of their respective companies.

PhUSE 2012

Annex 1: dataset *work_datdef* (extract)

DSORD	DATASET	DESCRIPTION	VARNUM	VARIABLE	LABEL	TYPE_	LENGTH	COMMENTS
1	ACQ	Asthma control questionnaire	1	BATCH	SITE CODE	num	4	
1	ACQ	Asthma control questionnaire	2	PNO	PATIENT CODE	num	4	
1	ACQ	Asthma control questionnaire	3	PROT	PROTOCOL CODE	char	15	
1	ACQ	Asthma control questionnaire	5	CQ_DATE	ACQ QUESTIONNAIRE DATE OF MEASUREMENT	char	7	Original variable V_DT from Questionnaire datasets
1	ACQ	Asthma control questionnaire	6	CQ_VIS	VISIT NUMBER	num	8	Derived from CQ_DATE
1	ACQ	Asthma control questionnaire	7	CQ_VISD	VISIT DESCRIPTION	char	40	Derived from CQ_VIS
1	ACQ	Asthma control questionnaire	12	CQ_Q1N	Q1 WOKEN BY ASTHMA FREQUENCY	num	8	Derived from variables QC_01A to QC_01G of Questionnaire datasets
1	ACQ	Asthma control questionnaire	15	CQ_Q2N	Q2 ASTHMA AT WOKE UP	num	8	Derived from variables QC_02A to QC_02G of Questionnaire datasets
2	AES	Adver Events	1	BATCH	SITE CODE	num	4	
2	AES	Adver Events	2	PNO	PATIENT CODE	num	4	
2	AES	Adver Events	3	PROT	PROTOCOL CODE	char	15	
2	AES	Adver Events	4	AE_SEQ	AE SEQUENCE	char	4	In order on onset date & ae preferred term
2	AES	Adver Events	5	AE_DATE	AE DATE ONSET	char	7	
2	AES	Adver Events	6	AE_TERM	AE TERM	Char	200	

(.....)