Create Descriptive Documentation with Hyper-links for SAS® Database

Don (Dongguang) Li, NCIC-CTG, Queen’s University, Kingston, Canada

ABSTRACT
Creating a descriptive documentation for SAS database is a common practice in all kinds of researches. The documentation will serve like a user’s manual of a production, providing full information about the database – how many data sets, how many variables in each data set, what are the attributes of the variables, how many observations in each data set, and what do those shortened variable names really mean. We all know that SAS CONTENTS procedure is designed for this purpose. It is easy to save the contents summary information to a data file that describes the database. Many times the database documentation will be shared with other data users when the database is transported externally. It is SAS professionals’ responsibility to provide an integrated document. People can do this in different ways. In this article, a useful program in plain SAS code is introduced, which can create a database summary table with hyper-links to the detailed descriptive tables of all data sets. The output documents are in both rtf and html formats.

INTRODUCTION
Whenever we have a SAS database, or even a data set, the first thing to do is get familiar with the data: what is the data about? What information does it contain? How many rows (observation) and columns (variables) in the data? What feature (type/length/format/informat) does each variable have? What do the names of the variables represent? To answer these questions, SAS programmers usually use PROC CONTENTS. However, do we really have to ‘proc contents’ the data all the time? Should there be documentation that describes the features of the database from the time it is born and can be regularly updated? I believe in practice the answer is yes at most of the places where research data are managed. Database documentation should be created along with the database. Whenever the database is used for analysis, or transported externally, or retrieved latterly for data mining / validation, or whatever will be done with the database, the documentation should always be along side the database.

Nobody will think it is a tough job to make a database descriptive document, and neither do I. What I would like to introduce is an automatic process (well, this is also easy) to make a summary table listing all the data sets with observation and variable numbers, and with hyper-links (this is the craft) to the details of the descriptive table for each data set.

PROGRAM LOGIC FLOW AND BASIC CODE
The program starts with the CONTENTS procedure for the defined database directory, and takes all the SAS data sets into a contents data file. Then a sub data set is extracted with only the data set names and data set-level information. This sub data set is used to create a database summary table. The detailed information of all data sets in the database will be reported in a table by memname (the variable that distinguishes the data sets). By producing the summary and details tables, an html link will be implanted with the variable ‘memname’ in the summary table and a corresponding anchor will be set with the by value in the detailed table. The chart below briefly describes the process:
/* The contents procedure outputs data contents of the entire data folder into a SAS data */
Libname dbdoc 'path of database folder'; %let trial=XX##;
proc contents data=dbdoc._all_ noprint out=content
details memtype=data;
run;

/* manipulate the output data to make a summary data and a detail data */
proc sort data=content(keep=memname name type length varnum label format
informat nobseq modate) out=dname;
by memname varnum;
run;
data varnum;
set dname(keep=memname nobseq varnum modate);
by memname varnum;
if last.memname;
rename varnum=varno;
run;
data dbdoc;
merge dname varnum;
by memname;
drop varnum;
run;

*** modify ODS template – omitted ***;

*** output the database information to document ***;
ods html body="dbdocsum.htm" style=bbackh;
ods rtf file="dbdocsum.rtf";

*** report summary table ***;
proc report data=varnum nowd headskip spacing=3 split='|' nocenter;
column memname varno nobs modate;

define memname / order width=15 left "Data Set Name";
define varno / display width=10 right "Number of|Variables";
define nobseq / display width=12 right "Number of|Observations";
define modate / display width=16 left "Last Modified|Date/Time";
compute before memname;
   mno + 1;
endcomp;
compute memname;
   urlstring="dbdoc.htm#db"||left(put(mno,2.0));
   call define (_col_,'url',urlstring);
endcomp;
title2 "SAS Database Documentation";
title3 "";
title4 "Introduction";
title5 "";
title6 "This document describes the statucture, contents and relations";
title7 "of all data sets in the database for trial %upcase(&trial).";
run;
ods rtf close;
ods html close;
*** Detailed report by data set (with html anchor defined) ***

options nobyline;

ods html body="dbdoc.htm" anchor="db1" style=.bbakh;
ods rtf file="dbdoc.rtf";
title;

proc report data=dbdoc nowd headskip spacing=3 split='|' nocenter;
column memname name type length format informat label varno nobs;
by memname;

define memname / order noprint;
define varno / order noprint;
define nobs / order noprint;
define name / display width=12 left "Variable|Name" ;
define type / display width=8 left "Var Type" format=typ. ;
define length / display width=6 right "Length";
define format / display width=12 left "Format";
define informat / display width=12 left "Informat";
define label / display width=25 left flow "Label of the variable";

break after memname / page;

compute after varno;
length var $100;
var=" Number of variables: 
  "||trim(left(put(varno,3.0))))||repeat(" ",70);
line var $100.;
endcomp;
compute after nobs;
length obs $100;
obs=" Number of observations: 
"||trim(left(put(nobs,5.0))))||repeat(" ",70);
line obs $100.;
endcomp;

title "Trial %upcase(&trial) - Data set: #byval(memname)"
run;
ods rtf close;
ods html close;

EXAMPLE
A hypothetical example is used to show the implementation of the program. Suppose we have a trial data folder that contains 12 data sets:

~\trials\data\ Tx1_bp
Tx1_con
Tx1_f1
Tx1_f3
Tx1_f5
Tx1_f6
Tx1_f9
Tx1_lab
Tx1_gol
Tx1_tox
Tx1eth
Tx1sum

The only modification needed is to define the folder with libname statement and assign a value to the macro variable ‘trial’ (%let trial=TX1 for the testing database). The CONTENTS procedure outputs contents of each data set into the output data. We expect one summary report along with a detailed report in 12 sections. The detailed table shows the features of all variables (name, type, length, format, informat, labels, as well as the variable and observation numbers) of all the data sets.

The hyper-links of the data set names with the corresponding detailed html tables are not only working for html summary table, but also for rtf table. In an effort to be concise, only the summary table in rtf format and one selected detailed table for a data set (CONMED) are displayed as below:
**DISCUSSION**

Using proc contents to get the descriptive information of the database is a well-known and commonly applied practice. This work is trying to integrate the processes into a one step program that automatically creates a database descriptive document with a database summary table and detailed tables for each data set. The program simultaneously produces the documentation in both rtf (for printer friendly) and html (for web friendly) formats. A technique is applied to make built-in hyper-links between the data name in the summary table and the corresponding detailed table of the data set. I hope this program may provide an easy way to make database documentation.

---

**REFERENCE**


**CONTACT INFORMATION**

Don (Dongguang) Li, PhD, MPH, Senior Biostatistician
National Cancer Institute of Canada - Clinical Trials Group
10 Stuart Street, Queen’s University, Kingston, Canada, K7L 3N6
Work Phone: 613 533 6000 Ext. 78337
Email: dli@ctg.queensu.ca