Create all those tables and call PROC SQL once!
Stanley Fogleman, Harvard Clinical Research Institute, Boston, MA

ABSTRACT
When creating extracts for clinical trial reporting, it is often the case that more than one dataset is created at a time. There is a great deal of overhead involved in calling or exiting PROC SQL (or many other SAS procedures, for that matter). The following presents a methodology for creating multiple datasets from one invocation of PROC SQL.

THE PROBLEM
In the course of creating a clinical trials reporting database, which might contain anywhere from 20 to 400 individual SAS datasets, the database on which the files reside need to be called numerous times. A great deal of overhead is used in the course of “setting up” and “knocking down” PROC SQL. One way to reduce this overhead is by telling the calling proc whether this is the FIRST, LAST, MIDDLE or ONLY call to the database.

ONE POSSIBLE SOLUTION
My approach was to “divide” up the PROC SQL calls as follows:

FIRST:

PROC SQL NOPRINT;
Connect to database (method-dependent)
Query

MIDDLE:
Query

LAST:
Query
Disconnect from database(method-dependent)
Quit;

ONLY: (combines all three above)
OMATIC: (causes step to be skipped)

So the SAS Supervisor just “sees” one call to PROC SQL, with many “create table” statements.

WHY BOTHER?
Well, if you are using pass-thru SQL, and have SAS/ACCESS licensed for the particular database you are using, and only have one or two calls to the database, it may not be worthwhile. But since the macro has been written as a shell, it only takes a few minutes to customize it to your particular needs.

IMPORTANT NOTE:
An “IN” macro which mimics the “IN” operator in BASE SAS is required for this macro to work correctly. SAS 9.1 has an %IN macro included (no pun intended). Since I am using SAS 8.2, I used David Ward’s %IN macro.

MACRO SMARTXTRACT
/**************************************************************************/
/* Import data from database tables into SAS using pass-through SQL */
/* odbc. */
/**************************************************************************/
%macro smartxtract(flag=,saslib=,sasmember=,
 saslabel=,instance=,schema=,tablename=,mergest=,wherepage=PAGENO,pageno=,visno=)
/ store des='transfer data from database to SAS';
%let context = CITY, STATE, ZIP;
%let sortord = CITY STATE ZIP;
%let instance = ABCD;
%let userid = EFGH;
%let pass = ijklnn;
%let conn = connect to odbc;
%let fromcon = select * from connection to odbc;
%let disconn = disconnect from odbc;
%let squote = %str(\');
%if %in(&flag,(FIRST ONLY))
  %then %do;
    proc sql noprint;
    &conn. (dsn="&instance." uid="&userid." pwd="&pass.");
    create table &saslib..&sasmember.(type=data label="&saslabel" sortedby=&sortord.) as
    &fromcon.
    (SELECT *
     FROM &schema..&tablename.&mergest.
     WHERE &wherepage. = &pageno.
     %if &visno ne %then %do;
     AND VISNO = &squote.&visno.&squote.
     %end;
     ORDER BY &context.);
    %end; /* first only */
%end; /* MIDDLE LAST only */
%if %in(&flag,(MIDDLE LAST))
  %then %do;
    create table &saslib..&sasmember.(type=data label="&saslabel" sortedby=&sortord.) as
    &fromcon.
    (SELECT *
     FROM &schema..&tablename.&mergest.
     WHERE &wherepage. = &pageno.
     %if &visno ne %then %do;
     AND VISNO = &squote.&visno.&squote.
     %end;
     ORDER BY &context.);
    %end; /* MIDDLE LAST only */
%end; /* LAST ONLY */
%if %in(&flag,(LAST ONLY))
  %then %do;
    &disconn.;
    quit;
    %end; /* last only */
%mend smartxtract;

DISCUSSION OF MACRO CALLING PARAMETERS

- Flag - indicates whether this is the FIRST MIDDLE ONLY or LAST call to the database. NOTE: Use of any string other than the ones listed above has the effect of “skipping” the step which may be useful in re-creating only certain tables
- Saslib - target SAS library
- Sasmember - target SAS member to be created,
- Saslabel - sas dataset label
- Instance - instance of DB
- Schema - schema of DB
- Tablename - host DB table name
- Mergest - optional appendage to table name
- Wherepage - optional selection criteria - defaults to PAGENO
- Pageno - page number selection criteria for host DB.
- Visno - visit number selection criteria - defaults to NULL.
DISCUSSION OF STATIC PARAMETERS
A thorny design decision was whether or not to include
certain parameters. The need for “information hiding” (such as as password and user-id information) led me to “hard-
code” the values inside the macro. Also, since we are typically just calling one type of database (Oracle, ODBC,
Access, for example), it was decided to include those values relevant to a particular database.

- context - comma separated values used in SELECT.
- sortord - space separated values used in SORTEDBY=.
- instance - internal DB identifier.
- userid - user-id for database.
- pass - password for database.
- conn - connection verbiage.
- fromcon - select verbiage;
- disconn - disconnect verbiage;
- squote - %str(%');

CONCLUSION
This macro provides a compact way to encapsulate the information needed for querying a database on multiple
occasions.

REFERENCES

ACKNOWLEDGMENTS
Brendan Gilbane, (1930-2001) former dean of College of General Studies, Boston University, for allowing a “C plus” student
into college.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Stanley Fogleman
Harvard Clinical Research Institute
930 Commonwealth Ave West
Boston MA 02215
Work Phone: (617) 632-1515
Email: vogelmann74@hotmail.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.