USING MACROS TO DEVELOP A FLEXIBLE END USER SYSTEM TO ACCESS, EDIT, AND PRODUCE REPORTS FROM ANY PERMANENT SAS® DATASET

Cynthia Miller, United Airlines

ABSTRACT

This paper will illustrate a menu driven system that allows an end user to access, edit and/or produce reports from any permanent SAS® dataset. Originally designed for end users who are totally unfamiliar with programming, use of the system has been expanded to include all types of users. Programmers without SAS® knowledge have found it helpful in accessing data stored in permanent SAS® datasets for import into other systems. SAS® programmers have found it to be a quick and easy method to use to access SAS® datasets without a great deal of coding.

Through macro driven menus, the user can (1) list the members and observations in a dataset; (2) view and/or edit the contents of a dataset via full screen edit; (3) delete a dataset member; (4) list the contents (variables) contained in a dataset; (5) copy a member of the dataset; or (6) produce a report from any permanent SAS® dataset. The report writing option uses macros to give the user flexibility in variable selection, subsetting options, sorts, titles, and labels.

INTRODUCTION

To those who are not familiar with the SAS® system, the permanent SAS® database can appear to be very complex. It cannot be used with ISPF or accessed through the use of standard TSO commands. In order to simplify the use of the permanent SAS® database for non-SAS® users, the following menu system was developed using base SAS® software and SAS/FSP®.

The application consists of two main parts: (1) the macro used to display the menu and execute the options, and (2) the program used to provide the user flexible reporting capability. The combination of these two programs provides the user with "friendly" and flexible access to any permanent SAS® dataset.

THE MENU

The first part of the system, the menu, is produced with %PUT and %INPUT calls stored in a SAS® macro.

```plaintext
>>> PERMANENT SAS DATASET HELP MENU <<<
1. LIST THE MEMBERS AND # OF OBSERVATIONS IN DATASET
2. LOOK AT ONE OF THE MEMBERS OF THE DATASET VIA FULL SCREEN EDIT
3. DELETE ONE OF THE MEMBERS OF THE DATASET
4. LIST THE CONTENTS (VARIABLES) IN A MEMBER
5. COPY A MEMBER OF THE DATASET
6. RUN A REPORT FROM A PERM SAS DATASET
ENTER SELECTION OR X TO EXIT
```

After the macro is identified as SASMENU, the options are specified and the screen is set to a MOD2 terminal operation. A macro variable (&DONE) is established and set to zero. This variable is then used in a %DO %WHILE statement in order to remain in the macro, thus returning to the menu after each option until the user chooses to exit.

```plaintext
%MACRO SASMENU;
OPTIONS NONOTES LS=88;
%TSO I OQMOD2;
%LET DONE=0;
%DO %WHILE(&DONE=0);
```

The menu is then displayed on the screen through the use of %PUT %STR statements, and the users input is read with a %INPUT statement.
The input is then read and if the choice was 1-5, the user is prompted to supply the dataset name and the dataset is allocated.

If the user chooses the first option — "LIST THE MEMBERS AND # OF OBSERVATIONS IN DATASET" — the screen is cleared and a PROC DATASETS with the NOFS option is used to display the list of members. The statement PUT used with a PUT PAUSE puts the three "s to the screen and then waits for the user to press 'enter' or any key before returning to the menu.

The second option — "LOOK AT ONE OF THE MEMBERS OF THE DATASET VIA FULL SCREEN EDIT" — prompts the user for the member name and uses PROC PEDIT to display the data. A standard default screen has been used, however, a customized screen containing instructions may be preferred for new users.

When choosing the third option — "DELETE ONE OF THE MEMBERS OF THE DATASET" — the user is asked to supply the name of the dataset member to be deleted. They are then questioned to verify that the correct member is being deleted, and if the response is 'Y' a PROC DATASETS statement with the DELETE option is used to delete the member.

Option four — "LIST THE CONTENTS (VARIABLES) IN A MEMBER" — again prompts the user to supply the member name and then uses PROC CONTENTS followed by PROC FSLIST to display the contents of the member on the screen.
The fifth option — "COPY A MEMBER OF THE DATASET" — prompts the user for the name of the member to be copied for the name of the new member. A data step is then used to copy the member into a new member.

```sas
%if %upcase(&CMDIN)=5
  %then %do;
    %ts0 closecrn;
    %put enter member name you wish to copy;
    %input omember;
    %put enter member name for copy of &OMEMBER;
    %input cmember;
    data dataset.&CMEMBER; set dataset.&OMEMBER;
    run;
  %end;
%end;
```

The sixth option — "RUN A REPORT FROM A PERMANENT SAS DATASET" — includes a SAS program named REPORT which is described in detail below. PROC FSTLIST is once again used to display the report to the screen.

```sas
%else
  %if %upcase(&CMDIN)=6
    %then %do;
      %inc pgm(report);
      proc fstlist doname=ft12f01 cc; run;
    %end;
  %end;
%end;
```

The final option ‘X’ is used to exit the menu. The macro will execute until &DONE is not equal to zero; therefore, in order to exit the macro, simply set &DONE to a value other than zero. The ENDSAS command is then issued to exit SAS.

```sas
%else
  %if %upcase(&CMDIN)=x
    %then %do;
      %let done=1;
      endsas;
    %end;
%end;
```

If the value of &CMDIN is not valid, a message is put to the screen prompting the user to re-enter. A %END statement is issued to close the %DO %WHILE statement, and the macro is closed with the %MEND statement.

```sas
%else
  %put command &CMDIN not recognized. please reenter;
%end;
%mend sasmenu;
```

### The Report Program

The report program allows the user to produce a report from any permanent SAS dataset. Again using %PUT and %INPUT statements, the user is prompted for the information necessary to run the report.

First, the global variables are set and the user is prompted to enter the dataset name and member to be referenced. The dataset is then allocated.

```sas
%global dname dmember when;
%put enter dataset name - i.e. sugi namefile;
%input dname;
%put enter member name you wish to;
%input dmember;
%ts0 allocate (dataset) da('&DNAME') shr reu;
```

Four macros are then included to be called based on the user's choices.

The first macro, KEEPVAR, will allow the user to include only specific values of the variables that will be included in the report. The macro variable %YESNO is set to 'Y' and used in a %DO %UNTIL statement in order to have the prompt repeat for possible subsetting options until the response input is 'N'.

```sas
%macro keepvar;
  %put enter the variable name followed by;
  %put the value you wish to include;
  %input kvar kval;
  %let yesno=y;
  %do %until(%upcase(&YESNO)=n);
    if &KVAR = &KVAL;
  %end;
%end;
'DO CLSCRN;

%PUT %STR( ) DO YOU WANT TO SUBSET YOUR DATA TO INCLUDE;
%PUT %STR( ) ANOTHER VARIABLE WITHIN THE CURRENT;
%PUT %STR( ) DATAT;
%PUT %STR( ) ;
%PUT %STR( ) ENTER Y OR N;
%INPUT YESNO;

%IF %UPCASE(YESNO) = Y %THEN %DO:
%PUT ENTER THE SUBSETTING VARIABLE NAME;
%PUT FOLLOWED BY THE VALUE YOU WISH TO;
%PUT INCLUDE;
%INPUT KVAR KVAL;
%END;

%END;
%MEND KEEPVAR;

The macro DELVAR is called if the user chooses to delete specific values of variables. Again the macro uses the macro variable %YESNO with a %DO %UNTIL statement to repeat the prompt for multiple deletions.

%MACRO DELVAR;

%PUT ENTER THE VARIABLE NAME FOLLOWED BY THE VALUE;
%PUT YOU WISH TO EXCLUDE;
%INPUT DVAR DVAL;

%LET YESNO = Y;
%DO %UNTIL(%UPCASE(YESNO)=N);
%IF &VAR = &DVAR %THEN DELETE;
%PUT DO YOU WANT TO EXCLUDE ANOTHER VARIABLE;
%PUT ENTER Y OR N;
%INPUT YESNO;
%IF %UPCASE(YESNO) = Y %THEN %DO:
%PUT ENTER THE VARIABLE NAME FOLLOWED BY THE;
%PUT VALUE YOU WISH TO EXCLUDE;
%INPUT DVAR DVAL;
%END;
%END;
%MEND DELVAR;

The macro KEEPDEL puts a menu to the screen to ask the user if they would like to include specific values for variables in their report. If they respond with a (0), the macro KEEPVAR is called to include only specific values of variables. If their choice is (2), the macro DELVAR is called to delete specified values of variables. The third choice, to include all values of variables, requires no action.

%MACRO KEEPDEL;

%TSD CLSCRN;

%PUT %STR( );
%PUT %STR( );
%PUT %STR( );
%PUT %STR( ) IF YOU WOULD LIKE TO INCLUDE ONLY SPECIFIC;
%PUT %STR( ) VALUES OF YOUR VARIABLES IN THE REPORT, YOU;
%PUT %STR( ) MAY INDICATE YOUR PREFERENCE BELOW:
%PUT %STR( );
%PUT %STR( ) 1. INCLUDE ONLY SPECIFIC VALUES;
%PUT %STR( );
%PUT %STR( );
%PUT %STR( ) 2. DELETE SPECIFIED VALUES;
%PUT %STR( );
%PUT %STR( );
%PUT %STR( ) 3. INCLUDE ALL VALUES OF VARIABLES;
%PUT %STR( );

%INPUT NUM;

%IF %UPCASE(NUM) = 1 %THEN %DO:
%KEEPVAR;
%END;

%IF %UPCASE(NUM) = 2 %THEN %DO:
%DELVAR;
%END;
%MEND KEEPDEL;

The final macro, ORDERC, will prompt the user for variables to be included on the report, labels for those variables, and the sort order for the report. The variable names are read in as macro variables &VAR1 - &VAR5. The labels are read as &VN1 - &VN5 and then recoded for use in the label statement. The recoded variables &VARN1 - &VARN5 resolve with an equal sign and the label that was input as &VN1 - &VN5 in quotes. The macro variable &L1 then resolves to: variable name = 'label'... thru the last variable input. If less than five variables are input, the values resolve to blanks and do not affect the statement. The variables used for the PROC SORT statement are read in as &VARA - &VARE; and again, if less than five are input, the result will resolve to blanks and not affect the statement.

%MACRO ORDERC;

%GLOBAL VAR1 VAR2 VAR3 VAR4 VAR5 VARN1 VARN2 VARN3 VARN4 VARN5 L1;
The actual program is very simple. A work file is copied from the permanent SAS® dataset. Then the variable WHEN is made containing the system date and a CALL SYMPUT statement is used to convert this to a macro variable (&WHEN) to be used in the title statement. The macros KEEPDEL and ORDERC are called and conditionally execute the macros KEEPVAR and DELVAR. The user is prompted for a title for their report. They are asked to put the title in quotes so that it can be read in as one macro variable (&TITLE).

Now that we have all of the necessary data for the report, the options are set and a PROC PRINT statement is used to print the report. The LABEL statement uses the macro variable &L1; the VAR statement uses the macro variables &VAR1 - &VAR5; and the TITLE statements use macro variables &DNAME, &DMEMBER, &WHEN and &TITLE. The program is then submitted with a RUN statement and the file is freed.

The result is a report that the user has created based on their input. The dataset name appears in the upper left hand corner, the run date and page number in the upper right hand corner. The title, labels, variables and order are per the specifications provided by the user.

**SUMMARY**

The application presented above illustrates only a few of the options that can be made available to provide the user flexibility in accessing and reporting from permanent SAS® datasets. Use of the SAS® macro language and SAS/FSP® have taken the user by the hand and provided virtually unlimited access to their data.

For Additional Information Contact:

Cindy Miller
United Airlines
22800 Davis Drive
Sterling, VA 22170
(703) 742-4251

SAS® and SAS/FSP® are registered trademarks of SAS Institute Inc., Cary, NC, USA.