USING MACRO-DRIVEN MENUS TO SELECTIVELY PROCESS SAS PROCEDURES

Michael Gutting, Southern California Edison Co.

ABSTRACT

When writing a SAS* program to be executed by an individual with little or no SAS experience, the interface should be as transparent as possible. SAS is an ideal language to use in the creation of many different kinds of "production" programs. The difficulty with using SAS is that in order for someone to take advantage of all of the interactive capabilities of SAS, the user needs to have a working knowledge of the language. This sometimes limits the usefulness of an otherwise very desirable program.

A more desirable situation is one in which the user of the program has the ability to enter SAS interactively via a TSO CLIST, selectively process a wide variety of data, view the interim output, choose to print or delete the output upon termination of the program and have the entire process be transparent to the user.

The method discussed in this paper allows a non-technical user to selectively execute SAS code based upon responses to menus created using SAS/FSP*. This paper details a simple example of the creation of the CLIST, menus, and the interface between the menus and the SAS macros.

INTRODUCTION

Southern California Edison Company has an obligation to our customers to provide them with service at the rate schedule which would be the most advantageous to them. In the past, this involved bookkeepers throughout the company calculating customers' bills for an entire year, sometimes using several different tariff schedules.

The need for an automated system was clear. SAS is an ideal language for this process, but none of the company's bookkeepers have any training in SAS. A method was devised which allows the bookkeepers to logon to TSO, and execute a CLIST which calls SAS interactively. The bookkeeper is passed from one menu to another based on answers given in prior menus. Using proc fsedit in conjunction with menus allows the bookkeepers to add new information (i.e. customer's consumption) and update the variables used in the various rate calculations. The resulting data was used by bookkeepers to determine at which rate schedule the customer was receiving the lowest bill. The bookkeepers had access to SAS, updated two SAS datasets, executed several macros conditionally and selected the variables which would appear in the proc print, without ever realizing that they had entered SAS.

CLIST

The TSO CLIST used to run the program is very simple. If the user specifies the debug option in the CLIST, several of the macro debugging tools were included in the options statement passed to SAS. If the user accepted the default, the options statement was set to nosource, and the user's entry to SAS was completely transparent.

Upon termination of the SAS program, the CLIST offers the user options for the output. If the user chooses to view the output, the output is sent to the screen. If the user wishes a print of the output, they are given the option of which printer they wish the output routed to and prompted for the number of copies requested. Please refer to FIGURE 1.

PROGRAM

The program is invoked by the CLIST. The program execution begins by creating a SAS dataset containing an arrayed variable (QUEST) which is set to missing.

Proc fsedit is used to create the master menu which drives the program. The SAS dataset created initially is called by proc fsedit and each of the arrayed variables is used to allow the user to specify which facets of the program they wish to have executed. The options allow for the editing of the factors used in the calculations, the editing and inputting of the customers' billing consumption, the actual execution of the program and the termination of the program. Each of these options, except the option to quit, calls a sub-menu which again prompts the user for information. The program will return the user to this menu until the user specifies the option to quit.

After leaving the main menu, the arrayed variables are checked for values of 'X'. A value of 'X' sets a macro variable to 'ON' and causes the execution of one or more macros.

The process of using an arrayed macro variable in conjunction with the menus created using proc fsedit is the basis for the entire program. It allows for the user to specify the conditional execution of any combination of macros. The user will return to this main menu until they specify the option to quit which causes the macro 'QUITT' to be executed. Please refer to FIGURE 2.

If the user specifies the option "EDIT RATE COMPONENTS" the macro 'EDITRATE' is executed. This macro invokes proc fsedit and allows for the user to change any or all of the components of SCE's tariff schedules.
Instructions regarding the use of the PF keys are included on each screen to aid the user. This editing process makes use of several screens to avoid confusion for the user.

Upon termination of this macro, control of the program returns to macro 'CONTROLT'. Please refer to FIGURE 3.

If the user specifies the second option, "INPUT CONSUMPTION", macro 'EDITSTG' is executed. This macro invokes proc fsedit and allows the user to input the data relative to the customer. All of the information required by the bookkeeper to identify the customer and provide the data necessary for the billing comparison to be calculated is included in this menu. Once again, the instructions for the PF keys are included on the screen.

Upon the termination of proc fsedit, the program returns to macro 'CONTROLT'. Please refer to FIGURE 4.

If the user specifies the third option, "RUN PROGRAM", macro 'RUNALL' is invoked. Macro 'RUNALL' contains two other macros, 'GETRATE' which allows the bookkeeper to select the tariffs to be included in the comparison, and macro 'RUNPROG' which actually causes the rate comparison to be performed and the resulting data to be output.

Macro 'GETRATE' creates a temporary SAS dataset containing one arrayed variable (getit). The variable is set to missing and the user is presented with a menu asking which rates they wish to have included in the comparison.

After leaving proc fsedit, the arrayed variable is checked for values of 'X' and the macro variables are conditionally redefined with the values of the various tariff schedules requested or left set to missing. This terminates macro 'GETRATE'. Please refer to FIGURE 5.

Macro 'RUNPROG' merges the customers individual data with the rate component data and performs the calculations necessary for the billing comparison. The customers bill is calculated on every tariff, not just the ones requested in the previous menu.

This macro also contains a formal report which details the rate components used during the comparison. This was included to allow for the user to audit the results of the program.

Throughout the execution of this macro the user is sent messages to the screen using SPUT statements. This seems to be very reassuring to the user as they can feel comfortable that the computer is actually performing the necessary steps and isn't "stuck".

This macro also performs the proc print which contains the actual billing comparison. The label statement includes values for all of the tariff schedules even though it would be very rare for all of the tariff schedules to be used for a given customer. The selection of the tariff schedules included in the printout is determined in the VAR statement. When the user selected the desired tariff the value of the macro variable was redefined from "missing" to a tariff schedule. Those left as missing are included in the VAR and SUM statements but have no effect on the execution of the proc. Please refer to FIGURE 6.

SUMMARY

This method of creating menus using the SAS macro language is a very powerful tool which can be adapted to a variety of applications. The Revenue Requirements department within Southern California Edison Company is using this approach in a variety of programs with great success.

For further information, the author can be contacted by writing to the following address.

Southern California Edison Company
Revenue Requirements Department
2244 Walnut Grove Avenue
Rosemead Ca. 91770

(818) 302-6865

* SAS and SAS/FSP are registered trademarks of SAS Institute Inc., Cary, NC, USA.
MACRO CONTROL T;
TSO ALLOC REU FOOO DAI 'REVREQ.RATECOMP.SASDATA' I;
DATA _NULL_; DROP I;
DO 1 TO 4;
IF QUEST1='X' THEN CALL SYMPUT('01','ON')
IF QUEST2='X' THEN CALL SYMPUT('02','ON')
IF QUEST3='X' THEN CALL SYMPUT('03','ON')
IF QUEST4='X' THEN CALL SYMPUT('04','ON')
RUN;
SET ASK;
RUN;
MEND CONTROL T;

%MACRO EDITRATE;
PROC FSEDIT DATA='ASK SCREEN'XX.SCINIT OPTION='l'; RUN;
OUTPUT;
RESETT;
%MEND EDITRATE;

%MACRO EDITSTG;
PROC FSEDIT DATA='XX.ONE' OPTION='l' SCREEN::XX.SCR1B;
RUN;
%MEND EDITSTG;

/* PROGRAM EXECUTION BEGINS HERE. MACRO CONTROL T IS CALLED BY THE CLIST. DATA ASK CONTAINS AN ARRAYED VARIABLE QUEST */
/* WHICH IS SET TO MISSING BEFORE IT IS CALLED BY PROC FSEDIT. */
/* VARIABLES LISTED BELOW ARE ALLOCATED TO FILE ASK FOR USE IN APPLICATION */
/* DATA _NULL_;
DO I TO 4;
IF QUESTI='X' THEN CALL SYMPUT('01','ON')
IF QUESTI='X' THEN CALL SYMPUT('02','ON')
RUN;
ZIF 801 = ON THEN EDITEDIT;
ZIF 802 = ON THEN EDITEDIT;
ZIF 803 = ON THEN SCREEN=1
ZIF 804 = ON THEN SCREEN=2
%MEND CONTRL;

/* DURING THE EDITING PROCESSES, THE USER IS ASKED TO PLACE AN X "EXT TO THE TASKS YOU WISH TO HAVE PERFORMED. */
/* EDIT EDIT COMPONENTS */
/* IF QUESTI='X' THEN CALL SYMPUT('01','ON') */
/* IF QUESTI='X' THEN CALL SYMPUT('02','ON') */
/* RUN REPORTS */
/* IF QUESTI='X' THEN CALL SYMPUT('03','ON') */
/* QUIT */
/* ZIF 801 = ON THEN EDITEDIT; */
/* ZIF 802 = ON THEN EDITEDIT; */
/* ZIF 803 = ON THEN SCREEN=1 */
/* ZIF 804 = ON THEN SCREEN=2 */

/* FIGURE 1. */
/* FIGURE 2. */
/* FIGURE 3. */
/* FIGURE 4. */

/* MACRO EDITSTG; */
/* PROC FSEDIT DATA='XX.ONE' OPTION='1' SCREEN='XX.SCINIT' RUN; */
/* MEND EDITSTG; */

/* PROC FSEDIT DATA='ASK SCREEN=XX.SCINIT OPTION=1' RUN; */
FIGURE 5.

```plaintext
/* THIS SECTION OF THE PROGRAM PRODUCES A FORMAL REPORT DETAILING */
/* ALL OF THE BILLING COMPONENTS USED IN THE CALCULATIONS. THIS IS */
/* A REQUIREMENT OF SCE AND ITS USER TO VALIDATE THE RESULTS. */

DATA SHOW;
SET XX.RATER;
/* FILE PRINT HEADER */
PUT PAGE_ 1 352 'EIS RATE COMPARISON PROGRAM' //
352 'LISTING OF BILLING COMPONENTS' //
352 'STILL WORKING ON THEM...' //
PROC SORT DATA=ONE;
BY ACCOUNT RATE 1;
RETURN;
/* THIS SECTION PRINTS THE ACTUAL BILLING COMPARISONS FOR THE */
/* THE REASONABLE UTILITY THEN FINDS ITEMS FROM THE REPORT */
/* IF ONE OF THE MACRO VARIABLES HAS NOT BEEN SET */
/* DURING THE SELECTION OF REQUESTED RATES, THE VALUE OF THAT */
/* VARIABLE IS MISSING, AND WILL NOT BE INCLUDED IN THE PROC */
/* THIS MACRO TAKES THE CUSTOMER CONSUMPTION AND Merges It */
/* WITH THE RATE COMPONENTS. MESSAGES ARE SENT TO THE USERS */
/* IF ONE OF THE MACRO VARIABLES HAS NOT BEEN SET */
/* DURING THE SELECTION OF REQUESTED RATES. THE VALUE OF THAT */
/* NEW PROCESSING. COMPARISONS ARE PERFORMED FOR ALL OF THE */
/* PRODUCED BY ACCOUNT RATE 1 */
```

FIGURE 6.