ABSTRACT

This paper presents an advanced application of FSEDIT under version 6. This paper shows how to:

* Link customized help screens, built with PROC BUILD, to an FSEDIT screen for both overall general help information as well as for individual entry field help.
* Create customized commands.
* Link a selection menu built with PROC PMENU with a FSEDIT screen.
* Test for entry errors and display appropriate error messages.
* Link together two FSEDIT screens to provide a tool for previewing output before printing.
* Link customized PF key settings for both FSEDIT and OUTPUT windows.

INTRODUCTION

A focus of the Electric Engineering Department is coordinating Capital Budget Projects. The tools needed to coordinate over 1100 projects include data entry, data tracking and gathering, and printing customized reports on an as-needed and monthly basis. SAS FSEDIT, version 6, was the tool chosen because it could provide all of these capabilities. SAS/FSEDIT was chosen for ease of maintenance, future needs, and compatibility issues in future releases of the SAS product. The new capability of having a compiled SCL program linked to the FSEDIT screen was the key to developing a robust FSEDIT application. A comprehensive system incorporating the FSEDIT product as the main driver was developed containing selection menus, customized commands, general and individual screen field help, and customized printing capabilities. Improvement in the FSEDIT product enables comprehensive verification of data entry fields with appropriate error messages. The selection menus were developed using PROC PMENU and are linked into the FSEDIT screen by the SCL associated with the screen. An undocumented "trick" that combines two SCL functions enables capturing the WHERE command then 1) storing the subset for the customized reports and 2) processing the WHERE command in the FSEDIT session. This application uses two FSEDIT screens linked together via MACRO code. This allows the user to specify a desired report in the first FSEDIT screen. In the second FSEDIT screen the user can view this report before printing it out.

SCREEN CONTROL LANGUAGE (SCL)

A compiled program associated with the FSEDIT screen enables you to create a very robust editing environment. This was not possible in FSEDIT before version 6. SCL can have five main sections: 1) FSEINIT, 2) INIT 3) MAIN, 4) TERM, 5) FSETERM. The FSEINIT section executes only once; upon entering the FSEDIT application. The INIT section executes each time a new observation is displayed. With CONTROL ALWAYS on, the MAIN section is executed by the enter key or PF-key. The TERM section executes every time you leave an observation to view the next one. The FSETERM section executes only when you leave the FSEDIT application. Warning: Do not use a password on the FSEDIT screen as a syntax error will lock you out from the SCL.

FSEINIT SECTION

The SCL excerpt below shows the contents of the FSEINIT section for this application.

000008 FSEINIT:
000010 CONTROL ALWAYS;
000012 CALL EXECCMDI('SETHELP SCREENS.MAIN.GENERAL.HELP');
000014 LENGTH WHERESTR $200 HEADING $200 SORTBY $200 FIELD $200 DEST $4;
000017 MSG = 'EDIT YELLOW FIELDS.';
000020 RC=PMENU('SCREENS.MAIN.options')
000022 * Link customized selection menu to this FSEDIT screen;
000023 RC=PMENU('SCREENS.MAIN.options')
000024 CALL EXECCMD('PMENU ON');
000026 FIELD='';
000027 SORTBY='';
000028 RETURN;

Each of the five sections starts with their appropriate name, semicolon, and end with RETURN. Line 10 turns on CONTROL ALWAYS which enables the MAIN section to process all keyboard entries. Line 12 associates a customized help screen to this FSEDIT screen. Any time you press the HELP PF-key or type HELP on the command line will display the customized general help screen, SCREENS.MAIN.GENERAL.HELP. For line 12 Screens is the data set library reference name, MAIN is the catalog name, and General.help is the member name in the catalog. Line 14 defines some SCL variables which are utilized for the customized commands and reports. Line 17 links in the selection menu developed using PROC PMENU to the FSEDIT screen. Line 20 issues a display manager command that toggles on the selection menu from the
The INIT section checks a variable’s value and if it is equal to 00, highlights the variable field in red, placing the cursor there, and writes a message below the command line or selection menu.

The above SCL code is the first part of the MAIN section. Lines 52, 55, 58, and 61 are instrumental in the processing done throughout this section. Line 52 assigns the SCL variable, TSTCMD, the first word typed on the command line. The delineator between words is a single blank. Line 55 assigns the SCL variable, VARNAME, the current location of the cursor. The SCL Variable TYPE on line 58 assigns the type of word that SCL variable TSTCMD represents. Line 61 assigns the SCL variable, CMD2, the second word typed on the command line. Line 67 starts the processing of the HELP command.

To receive individual help for an entry screen field move the cursor (Not the graphics cursor) to the entry field of interest and press the HELP PF-key. The select structure starting on line 69 determines which individual help screen to display. If no individual help screen is displayed then show the customized general help screen. After displaying the individual help screen the default help screen is reset to the customized general help.

The above section of SCL code shows the error checking abilities now available with FSEDIT. In line number 203, the MODIFIED function is an SCL function that indicates if a screen field has been changed or not. The SCL variables PROJ NO, WORKSECT, AND LOC WORK correspond to the names of the screen field variables. The function INDEX in line 206 is one of the common functions available with base SAS. You can use any of these functions with the SCL. The ALARM statement sounds the audible bell at the terminal. The CURSOR
The customized commands developed are for producing quick individual, customized reports. The customized commands enable the user to specify the first two title lines (HEADING and HEADING2), sort the listing (SORTBY), define which variables to print (FIELD), and subset the data(WHERE). The code for the HEADING command is shown above. The WHERE command is not an unique customized command. Lines 323 through 384 show an example of how to capture the WHERE command for customized reports and for processing the reserved WHERE command in the FSEDIT session. Using the special keyword SHOW after the customized commands allows the current string values associated with those commands to be displayed on the message line. Using the keyword HELP after any of the special commands gives individual help concerning that command.

000323 ******** WHERE COMMAND ;
000324 IF (WORD (1) = 'WHERE' OR
000325 WORD (1) = 'where') THEN
000326 DO;
000327 IF WORD (2) = 'SHOW' OR
000328 word(2)='show' OR
000329 WORD(2) = 'HELP' OR
000330 WORD(2)='help' THEN
000331 DO;
000332 IF WORD(2) = 'SHOW' OR
000333 word(2)='show' THEN do;
000334 MSG =WHERESTR;
000335 CALL NEXTCMD();
000336 CALL NEXTCMD(); CALL NEXTCMD();
000337 ELSE DO;
000338 IF WHERESTR =" THEN
000339 IF WHERESTR =" AND (CMD2='also' OR cmd2='ALSO'1 then
000340 wherestr=wherestr |
000341 ' &' |
000342 IF WORD(2) =" THEN
000343 wherestr=' ';
000344 MSG ='WHERESUBSETTING STRING
000345 '-11-'
000346 CALL EXECCMDI ('WHERE' , 'NOEXEC');
000347 END;
000348 ELSE DO;
000349 IF WHERESTR =" THEN
000350 MSG =WHERE
000351 COMMAND HAS BEEN SPECIFIED';
000352 ELSE
000353 if wherestr =" and
000354 (cmd2='also' or cmd2='ALSO') then
000355 wherestr=wherestr |
000356 &';
000357 IF WORD(2) =" THEN
000358 wherestr=' ';
000359 ELSE DO;
000360 IF WHERESTR =" THEN
000361 WHERESTR=WHERESTR |
000362 "-11-'
000363 CALL EXECCMDI ('WHERE', 'NOEXEC');
000364 ELSE
000365 IF WHERESTR =" THEN
000366 MSG =WHERE
000367 COMMAND HAS BEEN SPECIFIED';
000368 ELSE
000369 MSG =WHERE
The above SCL code from the MAIN section processes the WHERE command. This code is unique in that the reserved command WHERE uses both customized processing and normal FSEDIT processing. Lines 323 through 341 are used to process the HELP and SHOW special second commands as previously stated. Starting with line 343, the WHERE command is processed. The SCL variable WHERESTR is a string variable that will contain the correct WHERE statement syntax for use with "PROC PRINT" or "PROC FSEDIT PRINTALL". The SCL variable WHERESTR is continuously appended to when specifying each time the WHERE command. Both the FSEDIT WHERE processing and the SCL variable WHERESTR are reset when entering just the WHERE command on the command line. Line 377 and 381 demonstrates the two undocumented "tricks" to capture a reserved command and process it normally. The function WORDTYPE will return to you the type of any one of the first three tokens from the command line just like the WORD function returns to you the value of any one of the first three tokens. The syntax for the WORDTYPE function is like the WORD function and is shown on line 377. Values returned from the WORDTYPE function include the following:

SEMI - the semicolon. To distinguish between a; ending a command and a ': value being searched for.

NAME - a SAS name value (like the name of a variable or data set).

Starts with a letter or underscore and contains only letters, numbers, or underscores.

STRING - anything the user places quotes around.

EOD - End of data. Tells us the command didn’t end with a semicolon but there are no other tokens on the command line.

NUMBER - a numeric value (possibly containing a ',', '-', or 'E').

INTEGER - a set of digits. Like a number with no special characters.

SPECIAL - operators used in expressions such as '=' or '>'.

There is a parameter you can place in the EXECCMDI routine that executes the specified command without executing statements in the MAIN section of your SCL program. This is the NOEXEC parameter as illustrated on line 381. To capture the command from the command line and rebuild it in release 6, you can take advantage of the WORDTYPE SCL function and the NOEXEC option for the EXECCMDI SCL routine as illustrated in the above SCL code, lines 352 through 384.
The customized command PRINTNOW, prints the data, outputting reports to the OUTPUT window. The block of SCL code processing the PRINTNOW command starts at line 508 and ends at line 624. Just like the other customized commands, you can get specific user help by typing the word HELP after the PRINTNOW command on the command line. This code to display the help screen is on lines 512 through 517. All other customized command specifications must have been done before issuing the PRINTNOW command. The values of the SCL variables associated with the customized commands (WHERESTR, SORTBY, HEADING, HEADING2, FIELD) are used within the processing of the PRINTNOW command, lines 508 to 624. The substitution of these values is done in this section of the code so that all of the other customized commands must have already been specified. Note the order the SCL processing is done for proper results of the report in the OUTPUT window. The SUBMIT CONTINUE-ENDSUBMIT blocks act as a FILO (First In Last Out) Que that releases the code in these blocks for execution in the reverse order they are submitted.
The second FSEDIT screen uses the above SCL MAIN section code, lines 38 through 140. The purpose of this FSEDIT screen is to allow you a chance to preview the output specified in the first FSEDIT screen. This screen uses customized PMENU selection menus. When specifying a printer, the user can specify either to set the laser printer to print in landscape or portrait modes. Lines 60 through 68 set the necessary parameters to have the printer set to landscape mode. Lines 71 to 79 set the parameters to have the printer set to portrait mode. After you view the output and indicate a printer location, the print option can be selected. Lines 91 through 137 sends the contents of the OUTPUT window to the desired printer in the desired format.

MACRO CODE

The two separate FSEDIT screens described when linked through the above macro code provides a method for the user to change, delete, add any member of the MAIN catalog. The customized keys member, .KEYS, should be in the same catalog as the FSEDIT screen.

REFERENCES


SAS, SAS/FSP, and SAS/FSEDIT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.