USING PULL-DOWN MENUS WITH SAS/AF™ APPLICATIONS

Joann High, Tennessee Valley Authority

Abstract

Release 6.06 of the SAS system provides a new facility for entering commands into the SAS/AF application, the PMENU procedure. In previous versions of SAS, the AF applications commands could be issued from either the command line or by using function keys. The command line allowed commands to be entered and then executed by pressing the Enter key. Function keys allow the command to be assigned to a key, thus simplifying how commands are issued and minimizing key entry errors.

This paper is intended to give the reader one of many possible solutions for designing an AF application system that uses function keys for navigation. After reviewing the SAS methodology presented below, the PMENU solutions should reduce the PMENU learning curve and aid in resolving syntax questions for others wishing to use these functions.

Introduction To The Problem

While designing a SAS/AF application system that is user friendly, it becomes obvious that the designer needs an additional tool besides the command line to increase the user-friendliness and to provide more control of the system. The term 'user friendly" not only means easy-to-use but that the system must always be in control of the program flow. Part of the system requirement for the SAS/AF application described below is to allow the system to be function-key driven. The problem then becomes how to remind the user what function keys are available.

Implementation Of The Solution

The PMENU procedure and associated software comprise a sophisticated menuing system that replaces the command line with an action-bar which lists items available to the user for selection. When an item is selected, SAS can either process the command, display information, display another list of selections in a column format (pull down menu), or prompt for more information (dialog box). With the PMENU procedure, the user can implement a customized action-bar and list the function keys available. This provides both the display of the function keys and the PMENU functionality of moving the cursor to an item on the action bar and pressing enter to execute the item.

The discussion that follows highlights invoking the PMENU procedure, creating a customized PMENU that displays and processes function keys, and several methods of invoking a custom PMENU within an AF application. The examples below also show PROC BUILD windows, the SAS procedure for creating AF applications, and the use of Screen Control Language (SCL) commands which are part of the SAS/AF application programming language. For the examples below the file catalog has been previously created and the library name assigned. The examples use the library name 'mylib' with the catalog name 'mycat'. The operating system for the examples described below is CMS. Changes for catalog and external file definitions will be required for other operating systems.

Invoking Default PMENUS

At some SAS installations the PMENU is the default while at other installations PMENU's are optionally installed. The SAS system has a default set of PMENUS for applications development. In SAS, PMENU can be invoked by entering PMENU ON at the command line. The command line is then replaced with the default PMENU action-bar. The functionality of the SAS default PMENU can be used to develop customized PMENUs for SAS applications systems.

To invoke the PMENUs during a program, the following Screen Control Language (SCL) command is issued (see SAS Technical Report P-199 for 6.06 SCL):

    CALL EXECCMDI ('PMENU ON');

And to disable the PMENU procedure, enter:

    CALL EXECCMDI ('PMENU OFF');

In SAS/AF applications, some SCL commands have their own default PMENU. For example, the following program fragment shows a SCL command DATALISTC that creates a window for displaying all the character data for the variable STATES from the opened data set DSID. (Note: the literal string in the DATALISTC command that is displayed as an instruction in the screen below; DATALISTC is for character data only).

```
CALL EXECCMDI ('PMENU ON');   /* turn PMENUS on*/
CALL WREGION (5,6,23,65);      /* define window size */

/* define list variable and window title */
RETURNC = DATALISTC (DSID, 'STATES',
 'Position the cursor on the desired option
 and press enter.');

CALL EXECCMDI ('PMENU OFF');   /* turn PMENUS off*/
```

If the PMENUs were not invoked, the command line would be displayed. Since PMENUs were invoked, the command line is removed and the DATALISTC PMENU is displayed at the bottom of the window. As one of the functions of DATALISTC, the PMENU allows the user to navigate through the list of character variables. The following screen would be displayed for the above commands:

```
+-Select Data------------------------------------------------+
|                                                            |
| Position the cursor on the desired option and press enter. |
|                                                            |
|      Alabama                                               |
|      Alaska                                                |
|      Arizona                                               |
|      Arkansas                                              |
|      California                                            |
|      Colorado                                              |
|      Connecticut                                           |
|      Delaware                                              |
|      Florida                                               |
|      Georgia                                               |
|                                                            |
|    Find          OK          Cancel        Help           |
+------------------------------------------------------------+
```

Creating Custom PMENUS

The example that follows is an AF application that creates a customized PMENU with an action-bar with 5 items (ITEM). The PMENU is named MENU1 by the MENU statement. For each item, the information in single quotes will be displayed. If the displayed information is an executable system command, such as ITEM 'END', the SELECTION option of the ITEM statement can be omitted. The example below is not an executable statement because the text string is 'PF3=END' and not the executable command 'END'. The SELECTION option from the ITEM statement links the ITEM information to a SELECTION statement that contains the command that is to be executed. This linkage allows the action-bar ITEM name syntax to be free form. The information in single quotes on the SELECTION statement is executable. The program that follows should be created as an AF application program (using PROC BUILD). The program below is called MENU1.PROGRAM. After a successful compile, submit the program from the SAS command line (outside the AF PROC BUILD procedure).

```
INIT:
    SUBMIT IMMEDIATE;
    PROC PMENU C=mylib.mycat;
    MENU MENU1;
        ITEM 'PF3=END'        SELECTION=PF3;
        ITEM 'PF7=PAGEUP'     SELECTION=PF7;
        ITEM 'PF8=PAGEDOWN'   SELECTION=PF8;
        ITEM 'PF10=LEFT'      SELECTION=PF10;
        ITEM 'PF11=RIGHT'     SELECTION=PF11;
        SELECTION PF3  'END';
        SELECTION PF7  'FORWARD';
        SELECTION PF8  'BACKWARD';
        SELECTION PF10 'LEFT';
        SELECTION PF11 'RIGHT';
    RUN;
    QUIT;
    ENDSUBMIT;
    RETURN;


MAIN:
    RETURN;


TERM:
    RETURN;
```

This program, MENU1.PROGRAM, will create a custom PMENU entry in the AF application library and store the entry as MENU1.PMENU. This custom PMENU can be called into an AF application program. Several examples of using custom PMENUs are described below.

Using the Customized PMENUS from the SAS/AF GATTR Window

To display a custom PMENU from the AF application, the PMENU catalog entry must be associated with the application. If the custom PMENU is to be used throughout the execution of the application screen, it can be defined in the GATTR menu window of the AF application program (see SAS/AF software manual for

Version 6.06). On the GATTR window, place the name of the custom PMENU, MENU1, in the COMMAND MENU field.

```
+-BUILD: GATTR SAMPLE.PROGRAM (E)--------------------------------+
|                                                               |
| WINDOW      NAME:  Custom PMENU Program                        |
|          START ROW: _         COL: _                          |
|       NUMBER OF ROWS: _        COLS: _    BANNER: COMMAND SELECT NONE |
|                                                               |
|   GENERAL      HELP:  ........  ........                       |
|                KEYS:  ........                                 |
|     LOOKUP DATA SET:  ........                                 |
|======> COMMAND MENU:  MENU1                                    |
|   PROMPT CHARACTER:  ?                                         |
|     SYSTEM OPTIONS:  NO EXIT   RESIDENT   EXTENDED TABLE       |
|                                                               |
| PARENT  NAME: ........      LIBREF: *                          |
|         TYPE: ........      CATALOG: *                         |
|                                                               |
|   WINDOW TYPE    STANDARD   DIALOG   DIALOG BOX    LIST   HELP |
|                                                               |
|   SCROLL BARS    HORIZONTAL    VERTICAL                        |
|                                                               |
+---------------------------------------------------------------+
```

The entry MENU1 links the PMENU entry (of the same catalog) to the AF application program. When PMENU is invoked in the AF program, the custom PMENU named MENU1 will be displayed instead of the default PMENU. To invoke the PMENU, place the PMENU ON/OFF commands in the INIT and TERM sections of the AF program as shown in this program fragment:

```
INIT:
    CALL EXECCMDI ('PMENU ON');
    RETURN;

MAIN:
    .... (other application commands can
         be here)
    RETURN;

TERM:
    CALL EXECCMDI ('PMENU OFF');
    RETURN;
```

This will display the custom PMENU MENU1 action-bar items in the default location of the screen, the top left corner.

Note: The placement of the action-bar can by altered be changing the AF GATTR Window Type to DIALOG. This will place the action-bar across the page bottom and display each item as a "button" with white letters and a solid yellow background. Using the Dialog Window type is especially useful when using the SAS/AF Extended Dynamic Tables (see SAS Technical Report P-199 Release 6.06).

Using the Customized PMENUS with FSLIST within an SAS/AF Application

FSLIST can be called from AF screens to display a non-SAS data set, such as a report layout. If the user needs to page right, left, up and down. Functions keys can be assigned to perform the functions. PMENU can be used to display these function keys on the FSLIST screen in the top left corner of the displayed screen. This provides the user with both the display of the function keys and the PMENU functionality of moving the cursor to a function key displayed on the PMENU action bar and pressing enter. A customized PMENU can be created in the same manner as the

previous custom PMENU with the function names assigned to the keys. The following code:

```
CALL EXECCMDI ('PMENU ON');   /* turn PMENUs on */

          /* assign custom PMENU */
CALL EXECCMD ('SETMENU mylib.mycat
     MENU1.MENU');

          /* display file     */
CALL FSLIST  ('"SAMPLE SAS A"');
CALL EXECCMDI ('PMENU OFF');  /* turn PMENUs off */
```

would display the physical CMS filename SAMPLE SAS A with the custom PMENU named MENU1 displayed in place of the command line. It is important to note the following statements regarding this procedure:

  - PMENU ON/OFF commands are executed when control returns to the procedure,
  - SETMENU is executed after the PMENU ON command,
  - The SETMENU statement uses the fileref as MENU not PMENU, and
  - Since a CMS filename is being displayed, embedded quotes must also be used with the filename on the FSLIST command.

The following is an example screen of the custom PMENU action-bar MENU1 using the FSLIST statements above.

```
+-FSLIST: SAMPLE SAS A -------------------------------------+
! PF3=END  PF7=PAGEUP  PF8=PAGEDOWN  PF10=LEFT  PF11=RIGHT  !
!                                                           !
!             SUPER SHOPPERS MAILING LIST -- AS OF MARCH 1990!
!                                                           !
!       NAME              ADDRESS                   PHON!
!  --------------------   --------------------------  ------!
!  Allen, James R.        11003 Wolf Valley Road     690-45!
!  Armour, Sonia Rene     255 Meadowview Drive       691-81!
!  Barnes, Raymond S.     1217 Dogwood Road          694-12!
!  Bennett, Gary          2324 Church Avenue         690-22!
!  Boston, William Scott  308 S Pond Grove Drive     689-09!
!  Carter, John D         601 Gallaher Road          531-25!
!                                                           !
+----------------------------------------------------------+
```

Conclusion

PMENU is an outstanding new feature of the SAS system that can save hours/days of programming time and provide the programmer, designer, and user with a much more user-friendly system. PMENUs are extremely useful for aiding the performance of function key-driven application systems and for providing the needed application system control. In many instances, the default PMENU may contain all the functionality needed in the system. Alternatively, a set of custom procedures can be created to perform all the required functions. Once the basic concept and syntax for the PMENU procedure has been achieved, this concept can then be applied to other system application needs.

Author may be contacted at:
Tennessee Valley Authority
400 W. Summit Hill Drive   WT 7D-K
Knoxville, Tennessee   37902
(615) 632-8663