Real Men Do Use Menus: Developing Applications with SAS/AF® Software

Kenneth E. Johnson, Rand McNally and Company

Introduction

SAS/AF® software is a powerful application facility for developing user-friendly front ends to business applications. With SAS/AF software you can provide easy, menu-driven access to SAS® family of software products.

Perhaps the greatest benefit of SAS/AF applications is that they bring the power of SAS software to non-technical users. With SAS/AF, you can design menus and fill-in-the-blank SAS programs that can be used by end users with varying degrees of expertise with SAS software. Indeed, you can build an information system that can be used by those having no data processing experience whatsoever.

The proficient SAS user and the SAS programmer will find there are other benefits of using SAS/AF front-end SAS software. One standardized front end can be used for many business functions. Generic SAS programs can be developed that will handle many different tasks. The programs are only written once, with the end users supplying the information specific to their particular needs. SAS/AF also allows you to build custom computer-based-training (CBT) programs. Finally, programmers and analysts will find that they can quickly and easily develop application prototypes with SAS/AF.

SAS/AF Screens

A SAS/AF application includes a number of different screens:

- CBT presents training material, questions, and answers
- FORM gives specifications for printing to devices for SAS/FSP® procedures
- HELP displays help information to the user
- KEYS specifies PF key settings for full screen procedures
- LIST contains values used to validate user entry fields
- MENU contains menu choices; gives the user alternatives from which a selection is made
- PROGRAM contains SAS program statements to be submitted to the SAS system; can contain fill-in-the-blank fields that will be inserted into the program before submission

Procedures used with SAS/AF

- BUILD for creating or editing screens
- DISPLAY for running the application; displays CBT, HELP, MENU, and PROGRAM screens
- CIMPORT for converting library of screens into transportable sequential file format
- CPORT for converting CPORT sequential file into a library of screens
- CATOUT for writing information about a SAS/AF catalog to a SAS data set or external file

Screens created with PROC BUILD are stored in a SAS catalog, which is in turn stored in a SAS data library. The catalog has a two level name (libref.catalog) very much like a permanent SAS data set. However, the catalog name must be seven characters or less.

PROC BUILD

The syntax for PROC BUILD is:

PROC BUILD CAT=libref.catalog;

Other options include:
- MERGE=catalog — catalog to be copied to the catalog specified in CAT= option
- REPLACE — replace members in CAT= catalog with same named members in MERGE= catalog
- SELECT name.type — selects screen to be copied from MERGE= catalog to CAT= catalog; if not specified all screens are copied

When PROC BUILD is executed, a directory of all screen entries in the catalog is displayed. Information includes name, type, description, and last update date. The screen description, which will be displayed to the user, initially defaults to name.type. It can be changed by the DES command or R(ename) selection field command.

PROC DISPLAY

The Display procedure will display the screens built by PROC BUILD. The syntax is:

PROC DISPLAY CAT=libref.catalog.name.type;

libref refers the the SAS data library where the catalog is stored
catalog is the name of the specific catalog
name is the name of the screen to display
type is the screen type — CBT, HELP, MENU, PROGRAM

Menu Screens

Menu screens are the heart of a SAS/AF application. They simplify the user’s interaction with the application by providing a list of alternatives from which to select. Each selection will transfer control to another screen, where the user can perform various activities.

The transfer of control for each menu option is specified on the Menu Attribute Screen (see Fig 1). Each menu has a parent screen, which is the screen displayed when the user presses the END key from the menu. If no parent screen is specified, the user will return to the SAS system — which may be confusing for novice users. To prevent an exit out of the application the SAS system, you can code a main menu with itself defined as the parent screen. You would then code a menu option to exit the application; this option would pass control to a Program screen which would contain the ENDSAS; command.

The DISPLAY, CPORT, CIMPORT, and CATOUT procedures are included in Base SAS software with release 5.18.
The Attribute screen also specifies the screens to be processed as a result of each menu option. The Option column refers to an entry on the command line. The options default to numbers, but can be changed to letters by simply overtyping the number. The Key column refers to pressing a function key rather than entering the option on the command line. Key selections specified here override normal function key settings in effect. The Name and Type columns specify the screens to transfer control to for each option. Notice that the screens do not have to be in the same catalog as the menu; different catalogs are specified by filling in the libref and catalog fields.

Program Screens

Program screens are where the work of the application takes place, because they contain your SAS programs. Actually, the screens are made up of two parts: what the user sees, and what the SAS system sees. The two parts are separated by a Fill line; anything below this line is not seen by the user (see Fig 2 and Fig 3).

The first part of the Program screen is optional; it is only required if user responses are necessary to run the program. The user will fill in blanks with their responses and issue the END command to execute the SAS program. If there are no user responses the screen will begin with the Fill line, and the SAS program will be immediately executed when the screen is called.

Each program screen includes a PROC DISPLAY; to indicate where control is to be passed after the program has executed. You do not code parent screens for Program screens; rather you explicitly process another screen through programming statements.

User-Supplied Values

When designing a Program screen, user fields are denoted by an ampersand as the first character (i.e., &name). Each field must have a unique name. The values entered are substituted in the program below the Fill line before program execution. Note that these ampersand fields are a feature of SAS/AF software and are separate from the SAS macro facility.

User Field Attributes

The Program Attribute screen defines the attributes for all user fields on the screen (see Fig 4):

- **Field Name** — name of the field as specified in the top part of the screen. You name the field as &name plus as many underscores as necessary to reach the necessary field length. For single character fields specified by a single ampersand, the system will assign the default name of FIELDn where n is the nth field on the screen.

- **Length** — total length of the field, including ampersand and any underscores concatenated to the name

- **Row and Column** — location of the field on the screen

- **Type** — the type of the field:
  - **Name** — any valid SAS name
  - **Num** — standard SAS numeric field
  - **Char** — standard SAS character field
  - **Fixed** — integer numeric values
  - **Action** — one character field that contains either a blank or non-blank value; default for single character fields (FIELDn)

- **Input** — name of an existing SAS data set
- **Varlist | Varlist | Varlist** — field containing SAS variable names (Varlist means any variable, VarlistN indicates numeric variables, VarlistC indicates character variables); name of the SAS data set is specified in the List field

- **Caps** — all characters entered will be translated to uppercase

- **Just** — justification for the field (L, R, C)

- **Pad** — character used to fill the field when the user leaves the field blank

- **Protect** — protect the field from user entry

- **Non-Display** — field will not be displayed on the screen

- **Format and Informat** — any valid SAS format and informat

- **List** — indicates list of valid values for the field. There are several ways to indicate valid values:
  - Indicate list of valid values separated by blanks (e.g., M F)
  - Specify a range by entering a "c" followed by the range (e.g., <100)
  - Specify a LIST screen preceded by "#" (e.g., BUCC.LIST)
  - For a field with the VARLIST | VARLIST | VARLIST type, enter the name of the SAS data set containing the variables specified

- **Help** — name of a HELP or MENU screen to be displayed when the user positions the cursor on the field and requests help

- **Associated SAS Macro Variable** — allows you to specify a SAS macro variable, and the value entered by the user will be automatically associated with the macro variable. This feature allows you to save values across screens in the application, since the value of the macro variable remains after the particular Program screen has been submitted.

Conditional Execution

A powerful feature of Program screens is that they allow for conditional execution of SAS code depending on whether or not user values have been entered in a field (see Fig 5).

### fieldname tells the SAS system that if user field fieldname is left blank, the statements that follow are not to be executed. Conditional execution can be nested to one level; the lowest level is indicated by a single pound sign (#年年底). You can proceed ### fieldname or fieldname with a not sign (!) to indicate that the following code should be executed only if &fieldname is left blank.

The conditional execution remains in effect until another conditional execution indicator of the same or higher level is encountered (or end of the program). A # followed by a blank indicates unconditional execution of the code that follows; and ends the effect of #### or ##### control.
Help Screens

Help Screens are text screens which can be used to document or clarify items on Menu and Program screens. There is no limit to length of a Help screen; scrolling commands can be used to view the entire screen.

The Help screen can be associated with the overall screen or with specific user fields. When building a screen, use the `HELPNAME name.type` command to specify the Help screen for that screen. For help on specific user fields, specify `name.type` in the Help field on the Program Attribute screen.

List Screens

List Screens are used to define valid values for user fields on Program Screens. When creating the List screen, you specify the attributes for the List field: length, type (num or char), and optional format and informat. The List screen will then display a series of blanks into which you enter the valid values in any order. The system will sort the list automatically.

SAS/AF Hidden Tricks

Some feature of SAS/AF are not documented in the SAS/AF User's Guide. These are discussed in Technical Report: P-146, Changes and Enhancements to the Version 5 SAS System.

Consistency Checking

The List field on the Program Attribute screen can check a single user field, but there is no way to do consistency checks without submitting the SAS code below the FILL line. However, you can write a macro to examine multiple user fields and flag errors immediately, before the SAS program is submitted.

To create a consistency macro, code three pound signs in columns 1-3, followed by the macro, immediately below the FILL line. The macro can be written within the Program screen, or defined elsewhere. End the `###` macro by coding three pound signs in columns 1-3 on the line following the last line of the macro.

The `###` macro is invoked before the Program screen is displayed to the user, and each time the user presses the ENTER key or a PF key. A number of predefined SAS macro variables are available to control processing within your macro. For example, the `_DCALL` macro variable can be checked to determine if it is the initial display of the screen, if the user pressed the END key, or if the user issued the CANCEL command.

Writing to an External File

The `###` option allows you to write user field values to an external file. You must first associate a fileref with the external file. Code `###` in columns 1-3 below the FILL line (and after any `###` macro), followed by a space and the fileref where the information is to be written. You end the `###` option by coding `###` in columns 1-3 on the line following the last line to be written. Any text or code between the two `###` is written to the external file after user field replacement. Additional SAS code for execution can be included after the second `###`.

Prototype SAS/AF Applications

An excellent resource for SAS/AF developers are the prototype SAS/AF applications available from the SAS Institute. These prototypes are available free for releases 5.16 and 5.18 under OS, CMS, and VMS. They require the product featured to run; "optional" features require SAS/FSP, SAS/GRAPH, and SAS/AF. The prototypes include:

- SAS/AF Sample Menu System
- Report Writing Prototype
- Graphics Prototype
- Forecasting Prototype
- Operations Research Prototype
- Statistical Quality Control Prototype
- CPE Starter Set (Computer Performance Reporting)

The SAS/AF Sample Menu System is shipped as part of Base SAS Software release 5.18. The CPE Starter Set is available under OS only and is used in conjunction with Barry Merrill's MXG product.

Documenting SAS/AF Applications

SAS/AF software has only limited capability to document SAS/AF catalogs. Technical Report: P-146 Changes and Enhancements to the Version 5 SAS System includes a way to document the text portion of CST, Help, Menu, and Program screens.

PROCCBUILD CAT=libref.catalog
PRTFILB=fi1eref;

Unfortunately, this method does not provide information on List, Form, and Keys screens, or variables and attributes of the other screen types. Luckily there is an excellent program to document a complete SAS/AF application: AFLIST.

AFLIST was developed by R. David Sisemore, General Dynamics/Convair Division. It was first presented at SUGI 12 as a macro; a new stand-alone version was presented at SUGI 14.

No SAS/AF developer should be without AFLIST!

AFLIST provides:

- Cover Page
- Index to AF Screens, including name and type, screen description, page number, subtotal of screens by type, and grand total of screens
- Cross-reference listing of macro variables, both Variable-Screens listing and Screen-Variables listing
- Listing of all screens as they appear in PROC BUILD, including attribute screens
There are a number of ways you can use the prototype applications. They can demonstrate the capabilities of SAS software to potential users, particularly the easy access to the SAS system through the SAS/AF interface. The prototypes include sample data sets, and SAS demonstrations can be scheduled on short notice.

The prototypes are also a resource for SAS analysts and programmers. The prototype programs can be used as training material, since they show examples of menu-driven systems for non-technical end users, as well as examples of more complex SAS programs. In fact, the prototypes can be used as code libraries while developing your own SAS/AF applications. Of special note are the excellent Help screens, which can be reused in your own application.

However, there are some caveats when using the prototypes. The prototypes themselves can have slow execution times due to the complexity of some of the Program screens. The programs themselves are not well internally documented. Some options, such as adding a variable to a data set in the Report Writing Prototype, do not work and must be modified through PROC BUILD. Finally, quite often Help screens are not associated with Program Screens, even through "PRESS HELP FOR ASSISTANCE" appears on the user portion of the screen.

Sample Programs for the Prototype Applications:

- REPORTS1.PROGRAM
  - Creates list data set of data sets in library
  - Creates macro variables for data set names
- REPORTS2.PROGRAM
  - User selects type of report to print
  - Available data sets are shown to user on the screen, using macro variables created in Reports1.Program
  - Selected data set name to print is validated as Varlist field type
- VARINFO.PROGRAM
  - Creates list data set of variable names, list data set of numeric variables, list data set of character variables, for data set specified by user
  - Creates macro variables for variable names, numeric variables, character variables
- RPT4CHT.PROGRAM
  - Proc Print of variables from specified data set, with sum option
  - Data set variable names shown to user on the screen, using macro variables created in Varinfo.Program
- REPACHT.PROGRAM
  - Produces bar or pie chart
  - Uses SAS/GRAPH for high resolution graphics if user enters device name; user can enter graph options (e.g., fonts, colors)
  - Data set character and numeric variables shown to user on the screen, using macro variables created in Varinfo.Program
- UTILSUB.PROGRAM
  - Subsets a data set to create new data set
  - Available variables and their type shown to user on the screen
  - Uses message field to communicate with user; indicates if new data set was successfully created and if it has no observations
- REP407.HELP
  - Lists SAS/GRAPH fonts and gives recommendations for titles and footnotes
- REP410.HELP
  - Explains pattern values and lists colors for SAS/GRAPH

**SAS/AF In Version 6.06**

SAS/AF Software in Release 6.06 will have several new and powerful features for developing SAS/AF applications. Among these:

- Screen Control Language — functions and routines to control fields on the screen and manage screens within the application
- SAS/AF applications are run with the AF display manager command
- Screens are stored as compiled executable files, so the screen is compiled only once
- SAS/AF will include a source level AF debugger
- TEST Command — you will be able to test the flow of screens and look of the display from within Proc Build
- Menu link — the user will be able to move through a number of menu levels at once (similar to ISPF)

**Conclusion**

SAS/AF software is a powerful application facility for providing user-friendly, menu-driven access to the SAS system. SAS/AF applications can be used by people having little or no data processing experience; a SAS/AF application’s menus and fill-in-the-blank programs can bring the power of SAS software to everyone.

**References**


Technical Report: P-175, Changes and Enhancements to the SAS® System, Release 5.18, under OS and CMS.


**Contact Information**

Kenneth E. Johnson
ISO Information Center
Rand McNally & Company
8255 N. Central Park Ave.
Skokie, IL 60076
(708) 673-9100

SAS, SAS/AF, SAS/FSP, and SAS/GRAPH are registered trademarks of SAS Institute, Inc., Cary, NC, USA.
Figure 1 -- Menu with Menu Attribute Screen

ISD BUDGET SYSTEM MAIN MENU

Command ===> 

1 RECORD INITIALIZATION
2 NEW CYCLE PROCESSING MENU
3 DATA ENTRY MENU
4 REPORT MENU
C SPECIFY NEW CYCLE TO WORK ON
E END BUDGET SYSTEM AND RETURN TO TSO

PRESS PF1 OR PF13 FOR HELP

-----------------------------------------------

MAIN.MENU

Command ===> 

<table>
<thead>
<tr>
<th>PARENT SCREEN</th>
<th>NAME</th>
<th>MAIN</th>
<th>LIBREF</th>
<th>CATALOG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NAME</td>
<td>MAIN</td>
<td>LIBREF</td>
<td>CATALOG</td>
</tr>
<tr>
<td>OPTION</td>
<td>KEY</td>
<td>NAME</td>
<td>TYPE</td>
<td>LIBREF</td>
</tr>
<tr>
<td>1</td>
<td>___</td>
<td>RECINIT</td>
<td>MENU</td>
<td>*</td>
</tr>
<tr>
<td>2</td>
<td>___</td>
<td>SETUP</td>
<td>MENU</td>
<td>*</td>
</tr>
<tr>
<td>3</td>
<td>___</td>
<td>DE</td>
<td>MENU</td>
<td>*</td>
</tr>
<tr>
<td>4</td>
<td>___</td>
<td>FORMATS</td>
<td>PROGRAM</td>
<td>*</td>
</tr>
<tr>
<td>E</td>
<td>___</td>
<td>ENDSAS</td>
<td>PROGRAM</td>
<td>*</td>
</tr>
<tr>
<td>D</td>
<td>___</td>
<td>DATASET</td>
<td>PROGRAM</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>___</td>
<td>CYCLEYR</td>
<td>PROGRAM</td>
<td>*</td>
</tr>
</tbody>
</table>
CREATE EXPENSE RECORDS PROGRAM

Command ===> 

EXPENSE CATEGORY: ____________________________ (READ HELP IF FEES/DUES)

EXPENSE DETAIL: _____________________________ (IF NO DETAILS, RE-ENTER CATEGORY NAME)

EXPENSE TYPE: _ D = DIRECT
I = INDIRECT

PLACE AN "X" NEXT TO THE COST CENTER(S) TO WHICH EXPENSE APPLIES:

- 9270
- 9271
- 9272
- 9273
- 9275
- 9276
- 9277
- 9252 <=== DIRECT EXPENSES ONLY!

PRESS PF15 TO PROCESS THE RECORD. ENTER "CANCEL" ON COMMAND LINE TO EXIT WITHOUT CREATING NEW RECORD AND RETURN TO MENU.
CREATE EXPENSE RECORDS PROGRAM

Command ===> 

EXPENSE CATEGORY: &CATEGORY (READ HELP IF FEES/DUES) 

EXPENSE DETAIL: &DETAIL (IF NO DETAILS, RE-ENTER CATEGORY NAME) 

EXPENSE TYPE: & D = DIRECT 
I = INDIRECT 

PLACE AN "X" NEXT TO THE COST CENTER(S) TO WHICH EXPENSE APPLIES: 

& 9270 
& 9271 
& 9272 
& 9273 
& 9275 
& 9276 
& 9277 
& 9252 <== DIRECT EXPENSES ONLY! 

PRESS PF15 TO PROCESS THE RECORD. ENTER "CANCEL" ON COMMAND LINE TO EXIT WITHOUT CREATING NEW RECORD AND RETURN TO MENU. 

(Fill Line) 

DATA TEMP; 
SET TEMP; 
EXPCAT = ' &CATEGORY ' ; 
EXPDESC = ' &DETAIL ' ; 
EXPTYPE = ' &FIELD3 ' ; 
C9270 = ' &FIELD4 ' ; 
C9271 = ' &FIELD5 ' ; 
C9272 = ' &FIELD6 ' ; 
C9273 = ' &FIELD7 ' ; 
C9275 = ' &FIELD8 ' ; 
C9276 = ' &FIELD9 ' ; 
C9277 = ' &FIELD10 ' ; 
C9252 = ' &FIELD11 ' ; 
RUN; 
PROC APPEND BASE=WORK.INITEXPR DATA=WORK.TEMP; 
RUN; 
PROC DISPLAY C=BCAT.BUDGET.EXPREC.PROGRAM; 
RUN; 

233
Figure 4 -- Program Attribute Screen

ATTRIBUTES FOR PROGRAM & SYSTEM SCREENS

Command ===>

FIELD NAME: CATEGORY LENGTH: 25 ROW: 1 COLUMN: 26
  TYPE: CHAR  CAPS:X JUST:L PAD:_ PROTECT: NON-DISPLAY:
  FORMAT:  INFORMAT:  REQUIRED:X
  LIST: =VALCATS LIST
  HELP: FEESDUES HELP  ASSOCIATED SAS MACRO VARIABLE: ECAT

FIELD NAME: DETAIL LENGTH: 25 ROW: 3 COLUMN: 26
  TYPE: CHAR  CAPS:X JUST:L PAD:_ PROTECT: NON-DISPLAY:
  FORMAT:  INFORMAT:  REQUIRED:X
  LIST:  HELP:  ASSOCIATED SAS MACRO VARIABLE:

FIELD NAME: FIELD3 LENGTH: 1 ROW: 5 COLUMN: 26
  TYPE: CHAR  CAPS:X JUST:L PAD:_ PROTECT: NON-DISPLAY:
  FORMAT:  INFORMAT:  REQUIRED:X
  LIST: D I
  HELP:  ASSOCIATED SAS MACRO VARIABLE:

FIELD NAME: FIELD4 LENGTH: 1 ROW: 11 COLUMN: 26
  TYPE: CHAR  CAPS:X JUST:L PAD:_ PROTECT: NON-DISPLAY:
  FORMAT:  INFORMAT:  REQUIRED:X
  LIST:  HELP:  ASSOCIATED SAS MACRO VARIABLE:

FIELD NAME: FIELD5 LENGTH: 1 ROW: 12 COLUMN: 26
  TYPE: CHAR  CAPS:X JUST:L PAD:_ PROTECT: NON-DISPLAY:
  FORMAT:  INFORMAT:  REQUIRED:X
  LIST:  HELP:  ASSOCIATED SAS MACRO VARIABLE:
PRINT REPORT GROUPS / ALL REPORTS

PLACE A "X" NEXT TO THE REPORT GROUPS TO PRINT AND PRESS PF3/PF15

& RETURN TO REPORT MENU

&D2MSG & D2 INTRADEPARTMENTAL TRANSFER REPORT
&D5MSG & D5 ALLOCATION OUT SUMMARY REPORTS
&D6MSG & D6 ALLOCATION OUT WORKSHEETS
&D4MSG & D4 ALLOCATION OUT WORKSHEETS SUMMARY
&D7MSG & D7 ALLOCATIONS OUT SUMMARY BY BUSINESS GROUP
&D8MSG & D8 EXPENSE WORKSHEETS
&D9MSG & D9 DEPARTMENTAL ZERO BUILDS
&D0MSG & D10 CONSOLIDATED BUDGET
&D13MSG & D13 YELLOW SHEETS
&DALLMSG & PRINT ALL REPORTS

REPORTS WILL BE RELEASED TO THE PRINTER WHEN YOU LOG OFF OF TSO. PRINT JOB WILL BE UNDER THE SPG NUMBER YOU ARE CURRENTLY USING.

%LET RAST~; %LET PRTFLAG~Y;
##FIELD1
%LET D2MSG~;%LET D4MSG~;%LET D5MSG~;%LET D6MSG~;%LET D7MSG~;
%LET D8MSG~;%LET D9MSG~;%LET D0MSG~;%LET D13MSG~;%LET DALLMSG~;
%LET RAST~*;
PROC DISPLAY C=BCAT.BUDGET.REPORT.MENU;RUN;
##FIELD3
%D2MAC /* MACRO TO PRODUCE D2 REPORT */
%LET D2MSG = *PRINTED*;
##FIELD5
%D5MAC /* MACRO TO PRODUCE D5 REPORT */
%LET D5MSG = *PRINTED*;
##FIELD7
%D6MAC /* MACRO TO PRODUCE D7 REPORT */
%LET D6MSG = *PRINTED*;

(program continues)

##
&RAST PROC DISPLAY C=BCAT.BUDGET.PRTALL.PROGRAM;RUN;