The Role of the SAS MACRO Facility in Relation to the SAS/AF Software

Andrew J.L. Cary, Syntex Research

Abstract

This paper discusses the role of SAS* macros in SAS/AF*. It briefly summarizes the ways macros can be used and then describes the role they play in SAS/AF. The relative benefits and disadvantages of a menu-driven application developed using macros alone and developed using Macros with SAS/AF are compared in terms of programming effort and real costs.

Introduction

Macros were first introduced as a means of substituting a short 'name' for a longer block of SAS code. This was particularly useful for interactive SAS use because it allowed more efficient terminal use by making it unnecessary to retype repetitive code. In 1982 SAS released a more powerful macro language, one that allowed much more powerful macros to be developed. Input/output routines were added to the facility, allowing interactive users to pass arguments to macros and across jobsteps. It became possible with the new facility to write menu-driven systems to conditionally execute SAS statements. These systems allowed casual or novice users easier access to the powerful data analysis facilities of SAS without needing to learn SAS.

This power did not come without a price. Macros in the new macro language often were difficult to debug, with cryptic and often seemingly unrelated error messages. Difficulties arose when certain combinations of characters were desired, resulting in more and more specialized, subtle and complex macro functions to cope with them. The new macros required careful and often expert programming to be used efficiently and correctly. The expertise in the 'expert' system had to be built in, both in the programming and application.

At SUGI10, the SAS Institute announced a new product—SAS/AF. AF gives the SAS programmer a new tool to build menu-driven 'expert' systems, one which allows easy interfaces to be constructed by straight-forward means. A new method was given to conditionally execute SAS statements and to input data into macro variables.

This paper addresses what the role of the SAS macro is in SAS/AF. It will begin with a brief description of the operation of AF and how it can utilize macros, and continue with an example of the use of Macros in an AF system.

SAS/AF

AF is a SAS program product designed to make 'user-friendly' menu driven systems using SAS as the programming language. It is composed of four SAS PROCs; PROC BUILD; PROC DISPLAY; PROC CIMPORT; and PROC CEXPORT. PROC BUILD and PROC DISPLAY are the PROCs used to create and run AF systems. PROC CIMPORT and CEXPORT are used to migrate AF systems to and from 'mini' and 'mainframe' environments.

AF systems are composed of 'screens'. These screens are linked together to produce a 'system'. There are seven types of screens each of which is stored in a catalog. These seven types are Menu, Help, Program, CBT, List, Forms, and Keys screens. Menu, Help, Program, and CBT screens are used to construct AF systems. Keys, List, and Forms screens are 'service' screens used for specific tasks by AF.

Menu screens are used to direct system flow from one screen to another. PROC DISPLAY displays the text portion of the Menu screen and awaits a selection from the user. When the selection is made the procedure routes the system flow to the Program, Menu, Help, or CBT screen assigned to that selection in the menu screen.

Help screens are used to display information to the user. When PROC DISPLAY routes the system to a HELP SCREEN it displays the contents of the screen, pauses until the END key is pressed and returns to the 'parent' screen.
CBT screens are designed to assist Computer Based Training. They direct system flow from the CBT screen to other screens on the basis of responses entered by the user. They are in a sense a combination of the Menu, Program, and Help screens.

Program screens are used to input data, verify it, and submit and execute SAS code. When PROC DISPLAY invokes a Program screen, an optional data input screen is displayed. Data is entered into fields on this screen and into AF variables. The data is optionally verified against an informat and/or list and can also be placed into macro variables. After this the data is symbolically substituted into the screen’s SAS code and executed. Blocks of code are submitted based on the presence or absence of AF variables.

List screens contain a list of valid values which an AF variable can be validated against.

Forms screens are used to specify options for printing datasets.

Keys screens are used to define and store function key commands and values for screens.

PROC BUILD is used to define screens. It includes an editor and a screen catalog manager. When BUILD is invoked a catalog is identified to store screens in. The editor is then used to create and edit screens in the catalog. Screens are linked together to form AF systems. Figure 1 is a representation of a simple system used to allow a ‘naive’ user to subset, print, and plot a SAS dataset.

The Role of the Macro in SAS/AF

There are four basic uses of macros in SAS/AF:

1. Macro variables can be used to share data values between program screens. This allows for ‘default’ values for parameters.

2. Macro variables can be used to pass data from a datastep or from the system to program screens. This makes things like time-date stamps and data driven systems much easier to construct.

3. Macros can be used as ‘subroutines’ and ‘functions’. This traditional use of macros has been substantially improved by the autocall facility. Program screens can use macros the same as any SAS code can. This would allow a macro toolbox containing commonly used macros to be kept in one place and then invoked as needed.

4. Macros can be used to define new ‘Command-line’ commands for full screen operations. This is a new facility and provides a means of customizing the command line commands for a given application.

Examples of Macro Use

In order to demonstrate the role of macros in SAS/AF a simple example is in order. The example is laid out in schematic form in Figure 1 and in the outline for the system below. The example system consists of:

Menu(Main):
   Help(system use)

   1: Program(subset dataset)
      Help(dataset name)
      Help(subsetting)
      Help(toolbox tools)
      Help(variables)

   2: Program(Print dataset)
      Help(dataset name)
      Help(Titles)
      Help(Footnotes)
      Help(Sort)
      Help(By variables)
      Help(ID variables)
      Help(variables)

   3: Program(Plot dataset)
      Help(dataset name)
      Help(Titles)
      Help(Footnotes)
      Help(plot symbols)
      Help(axis values)
      Help(variables)

There is one menu screen, 3 program screens, and 11 help screens. Note that some of the Help screens are used in all three of the Program screens.

Macro variables are passed to and from the Program screens using an option of the Program screen’s Attribute screen. All AF variables are described on the Attribute screen and as a part of this description each AF variable can be assigned to a macro variable. Data can be shared between program screens by the simple expedient of assigning the same macro variable to AF variables on the same SAS code can. This would allow a macro toolbox containing commonly used macros to be kept in one place and then invoked as needed.

4. Macros can be used to define new ‘Command-line’ commands for full screen operations. This is a new facility and provides a means of customizing the command line commands for a given application.

Examples of Macro Use

In order to demonstrate the role of macros in SAS/AF a simple example is in order. The example is laid out in schematic form in Figure 1 and in the outline for the system below. The example system consists of:

Menu(Main):
   Help(system use)

   1: Program(subset dataset)
      Help(dataset name)
      Help(subsetting)
      Help(toolbox tools)
      Help(variables)

   2: Program(Print dataset)
      Help(dataset name)
      Help(Titles)
      Help(Footnotes)
      Help(Sort)
      Help(By variables)
      Help(ID variables)
      Help(variables)

   3: Program(Plot dataset)
      Help(dataset name)
      Help(Titles)
      Help(Footnotes)
      Help(plot symbols)
      Help(axis values)
      Help(variables)

There is one menu screen, 3 program screens, and 11 help screens. Note that some of the Help screens are used in all three of the Program screens.

Macro variables are passed to and from the Program screens using an option of the Program screen’s Attribute screen. All AF variables are described on the Attribute screen and as a part of this description each AF variable can be assigned to a macro variable. Data can be shared between program screens by the simple expedient of assigning the same macro variable to AF variables on
different Program screens. If the macro variable is defined prior to the invocation of the particular program screen, the macro variable's value will appear in that AF variable's data entry field. This feature allows the AF system programmer to supply default values to the data entry fields on a Program screen.

Data can be passed from the system in the same fashion, however 'read-only' macro system variables (ie. SYSTIME, SYSDATE) should not be assigned to an AF variable directly. This provides a means for the programmer to pass system information to AF data entry screens.

Example 1: Passing data from one program screen to another.

The dataset name in all three program screens is an AF variable called AF DSN. It is assigned to the macro variable M DSN on all three screens. Once the dataset name has been specified on one of the screens it will appear as the default value on the other two.

Command-line macros are a new feature to SAS and provide a means to add 'customized' commands to AF, FSP, and Display Manager command lines. These macros can be invoked from the autocall facility. These custom commands allow the AF programmer the opportunity to provide the user 'meaningful' commands for tasks such as redirecting output and issuing operating system commands.

Example 2: Command-line macro.

The Macro PrtDest prints a specified display manager stream to a specified-printer destination.

```
%MACRO PRTDEST(stream,dest)/CMD;
  X ALLOC FILE(XXYZZ)
  SYSOUT(A) DEST(&dest);
  PRINT &STREAM XXYZZ;
%MEND PRTDEST;
```

It is invoked on the command line thusly:

```
Command => %PRTDEST(OUTPUT,RMTO);
```

Macros used in the conventional fashion as self-sufficient 'subroutines' and 'functions' are used in exactly the same fashion as in ordinary SAS code. They are embedded into the program code associated with each program screen. A new and potentially very important use could be as pre-made tools called by the user in a DATA step to perform specified tasks by having the user select the tools from a menu. For example, a naive user could select a macro to log-transform a variable in a datastep by selecting a 'tool' and indicating the variable to use it on.

Example 3: The special transformation.

The subsetting Program screen has an optional 'toolbox' of transforms of the data that can be done. The following macro resides in the system 'toolbox':

```
%MACRO LOGIT(INVAR);
  LOG((1-%INVAR)/%INVAR)
%MEND LOGIT;
```

It was invoked when the SAS code was executed after the user entered the string below into the AF variable AF CODE1.

```
LOGITX = %LOGIT(X);
```

The same tool could be used in an expression in the AF variables data entry field:

```
LOGITR = %LOGIT(X) / %LOGIT(Y);
```

The Role of SAS/AF in the Macro Facility

The AF facility can provide a powerful tool as a data entry system for the macro facility in SAS. AP Program screens can provide many tools for the input and validation of macro variables. A classic example of this is the often convoluted coding that programmers go through to ensure that the user (a) specifies a real SAS dataset and that (b) given a 'real' SAS dataset, he selects a 'real' variable on it. AP provides an easy method for validating both the dataset and the variables in it. It also provides the means for validating data on input against a range or a list of values by specifying the range, list, or list screen in the list field of a program screen attribute screen.

The AF facility also provides the macro system designer an elegant and easy way to build menus and help facilities for data entry with a minimum programming effort.

Conclusions

One of the original goals of this paper was to compare the SAS/AF and the macro facility in terms of benefits and disadvantages in the construction of Menu driven applications. In the course of the
investigations leading to this goal it became obvious that the two approaches were in competition only in the actual 'Menu' portion of the application. It also became increasingly obvious that a comparison of the two approaches for building a menu-driven system was unreasonable. The AF facility is designed for this purpose with easy-to-use tools for constructing menus and screens that are lacking in the macro facility. The macro facility is designed as a more general tool, one much richer in control structures and traditional programming tools. The two facilities have a symbiotic relationship instead of a competitive one.

What was learned during the investigation? In short: If SAS/AF is available and a menu-driven system is needed, it is the method of choice. Powerful systems can be constructed by utilizing macros where appropriate in an AF system. AF provides powerful tools for validation, menu construction, and help facilities. Macros provide necessary data communication between the various elements of the system and between the system and its environment. They also can provide the means of giving AF a more powerful and flexible control structure.

* SAS and SAS/FSP are the registered trademarks of SAS Institute Inc., Cary, NC, USA. SAS/AF is a trademark of SAS Institute Inc.

REFERENCES


Example System

Example Menu
Command == >

1 Specify and Subset a database
2 Print specified dataset
3 Plot specified dataset

Select value and enter on command line. Press HELP key for help.

SUBSET
Program Screen
HELP Screen
HELP Screen

HELP
Help Screen

PRINT
Program Screen
HELP Screen
HELP Screen
HELP Screen

PLOT
Program Screen
HELP Screen
HELP Screen
HELP Screen
The Last Slide....

A terminal that says friendly things to you isn't necessarily user friendly....

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

SAS/AF is a trademark of the SAS Institute Inc.

IBM is a trademark of International Business Machines Inc.