Building User-Friendly Systems with Version 6
SAS/AF® and SAS/FSP® Software

Gail Kramer, SAS Institute Inc., Cary, NC
Linda Wharton, SAS Institute Inc., Cary, NC
K. Deva Kumar, SAS Institute Inc., Cary, NC

ABSTRACT

A goal of end-user applications is to be user-friendly. It is important that the applications programmer anticipate issues and questions that the user may have. The programmer must provide for a dialog between the user and the application so that decisions can be made while the application is in use. In Version 6 SAS/AF® and SAS/FSP® software, currently available in the PC environment, Screen Control Language (SCL) is used to design applications that can prompt the user for all the information needed to complete a task.

This tutorial expands on the ideas presented in the tutorial, "The Screen Control Language in Version 6 SAS/AF Software and SAS/FSP Software."

INTRODUCTION

There are many definitions of user-friendly used in the software industry today. Its meaning depends on the environment, the user, and the application. A complete and easily accessible on-line help facility may be sufficiently friendly in some applications. Pictures for point and press may be mandatory in others. In most cases, conversational prompting may be adequate. Despite this range of needs, it is possible to build in the appropriate level of interaction using Version 6 SAS/AF and SAS/FSP software.

The first section of this tutorial is an overview of the tools provided in Version 6 SAS/AF and SAS/FSP software that make building user-friendly interactive systems an easy task.

The second section of this tutorial is an introduction to the basic features of SCL and the default flow of control provided for interactive applications developed using Version 6 SAS/AF and SAS/FSP software.

The third section of this tutorial is an example of an interactive application. The example is designed to generate syntactically correct SAS procedure statements, enabling users without knowledge of SAS software to produce a report using the PRINT procedure.

TOOLS

SAS/AF software is designed specifically to facilitate building user-friendly interactive applications that are easily integrated with SAS/FSP procedures or other software products available in the SAS System. The basic tools available are described below.

**MENU** enables you to design a menu of items that can be executed based on a selection from the user.

**HELP** enables you to provide specific help information for an application, a screen, or a field.

**CBT** enables you to design sequences of text and questions to use for tutorials or computer-based training courses.

**PROGRAM** enables you to design screens to query the user for information and contains programs that are executed during the application.

These tools provide implementation support for creation of interactive applications at a basic level. This allows the developer to create as simple or as sophisticated a system as needed. The applications programmer can choose the appropriate tool based on the task to be accomplished. For instance, the creation of help for applications, individual tasks, or even specific fields on a screen is an operation that is readily accomplished. CBTs can be designed to provide additional levels of help or training that can be accessed at any time in an application. CBTs can also be used strictly to meet traditional computer-based training course objectives.

The basic building block of many applications is a menu. Menus make the integration of separate but related tasks an easy operation for the developer. Menus often provide the frame for an entire application because many applications are task oriented and a menu lends itself to listing tasks and branching appropriately. The task can be any of the basic tools that SAS/AF software offers:

- another menu
- a help
- a CBT
- a program.

A menu item selection can correspond to a single program, help, or CBT. A menu item selection can also provide easy access to a variety of related tasks by corresponding to another menu. If there are subtasks to be selected, you can provide intermediate menus before allowing selection of the final task.

If the task requires processing or prompting, the logical choice is a program. The program associated with a menu selection can range from short, simple SCL programs, designed to obtain necessary information or generate a few source statements, to complex control programs that manipulate an entire application.

A program provides additional tools for communication with the end user and enables the user to create a unique interactive process to be based on the application's specifications. The basic tools available for creating interactive dialogs in programs are described below.

**LEGEND** is a predefined window to display instructions or information.

**PREVIEW** is a predefined window to display generated SAS source code.

**DISPLAY** enables you to display additional windows within the current window.

**SELECTION** is a predefined window to display a list of SAS files and their contents from which the user can select one or more items.
SCL is a tool that enables you to create programs that control fields on the program screen and manage other screens in the application.

The LEGEND window can be used to provide instructions or to create a visual frame for the display of customized messages.

The SELECTION functions can be used to provide a list. This eliminates the need for a user to remember information necessary to accomplish certain tasks related to SAS files and their contents. For instance, a list of data sets related to an application can be displayed. Based on the data set selected, the variables in the data set can be displayed to prompt the user to select the variable to use in a given report or chart.

The DISPLAY capability can be used to create customized components for an application. This eliminates the need to collect all necessary information from one screen or to attempt to provide all information to the user on one screen. Visual prompts, user queries, and customized error handling can be broken down into single units. This means cleaner, less cluttered screens, avoiding confusion, frustration, and errors.

The tools listed above are just a few of the features available for producing a customized application. SCL is a well-defined programming language that allows iterative and conditional processing, SAS data set processing, user queries, customized error handling, screen management, and interaction with the SAS System.

GETTING STARTED

Applications should be implemented by

- designing the program or screen sequences
- linking programs or screens using menu, call DISPLAY, and conditional and iterative processing, where necessary
- implementing help.

You can design short, simple programs that generate only source code that is generated is enclosed in a submit block within the program. The submit block distinguishes SCL program statements from those programming statements submitted to the SAS System. A submit block begins with a SUBMIT statement and ends with an ENDSUBMIT statement. Within the submit block the developer can use any information collected in the entire application. Substitution is made for specified replacement strings when the source code is submitted for execution.

SAS/AF Flow of Execution

The default flow of execution for a SAS/AF SCL program is straightforward. When the user enters a program, the INIT label is executed. This is unconditionally done only once. Execution then pauses, waiting for the user to take some action. If a field is modified and the ENTER key or any other function key is pressed, the MAIN label is executed. This can be repeated any number of times. If a user chooses to execute the END or CANCEL command without modifying a field, the MAIN label is bypassed, and the TERM label is executed. By default, the TERM label is executed once before the user leaves the program.

SAS/AF Flow of Execution

Figure 1 SAS/AF Flow of Execution

FSEEDIT Flow of Execution

The flow of execution for an FSEEDIT SCL program needs to be broken up the execution into the two loops that can occur. Each time an observation is displayed, the INIT label is executed. Subsequent label execution depends on user action. If the user modifies the observation, both the MAIN and the TERM are executed before the user moves to the next observation. If the user does not modify the observation, neither the MAIN nor the TERM labels are executed.

The user can modify the observation any number of times, causing the MAIN label to be executed each time. The TERM label is executed only once, it occurs when the user exits the observation.

Now, the flow of execution, as it applies to the whole data set, can be defined. Two additional labels enable you to execute statements at the beginning and end of an FSEEDIT session. FSEINIT is executed once when the user enters the data set, and FSETERM is executed when the user exits the data set. These labels do not affect the action being taken on individual observations.
EXAMPLE

The example for this tutorial is based on the PRINT procedure, a base SAS® procedure. The goal was to develop an application that would allow a novice user to generate a report using combinations of options available with the PRINT procedure. In this example, the programmer chose to illustrate graphically a report and to update that illustration constantly as the user chooses options, changes options, or removes options. The procedure options that the programmer made available are presented to the user in the format of a point and press selection list. The Sel program behind the report illustration and options list controls the display of the illustration, controls the display of additional screens, and generates SAS source code. The additional screens are used only when the user requests an option that requires some specific information before the correct SAS source code can be generated.

Figure 3 Report Screen

In the GETTING STARTED section the default flow of execution for an SCL program was discussed. There are ways to modify the default flow of execution in an SCL program. The example described here uses the SCL CONTROL statement to indicate that the MAIN label is to be executed every time the user presses the ENTER key or executes a command. The syntax of this CONTROL statement is

CONTROL ENTER;

This one statement enables the programmer to set up the application to run off single keystrokes instead of depending on the user to modify a screen field. This implementation is friendly because the user is not required to know any commands and can accomplish any task using only the TAB and ENTER keys.

Figure 4 SAS/AF Modified Flow of Execution

Execution of statements in the MAIN label of this program is based on the word the cursor is on when the ENTER key is pressed. The SCL tool used to accomplish this task is the CURWORD function. This function simply returns to the developer the word on which the cursor is sitting when the ENTER key is pressed. The syntax for the CURWORD function is

word=CURWORD();

Conditional processing, which is dependent on the value under the cursor, is accomplished using the SAS SELECT statement, for example,

select (word);
  when ('SPACING') then do;
    call display('double .program');
    ds=symgetc('double');
    if ds='Y' then
      link dspac:e;
    else
      link sspac:e;
    cursor spacing;
  end;
  when ('LABELS') then do;
    more SAS statements
  end;
end;

1315
Perform task

END or CANCEL

Submit to SAS

Generate report

Get a Data Set

Data Set w/ ?

Curate to desired task

Press ENTER

Figure 5 Example Flow Chart

A data set selection list and a legend containing instructions on how to choose a data set from the list are the first items displayed for the user when the report writer task has been selected. In this example, an SCL-provided SELECTION tool, the DIRLIST function, was chosen to enable the user to select the data set to be printed. The syntax for the DIRLIST function is

\[ \text{dataset}=\text{DIRLIST}(	ext{libname, type, numsel, prefix}); \]

Using this function, the programmer can preselect the SAS data library, limit the selection to one data set, and append the library name to the data set name. Other Sel selection tools, such as VARLIST, are used with LEGEND windows when possible.

Figure 6 LEGEND and Data Set Selection Screen

The SCL DISPLAY function is also used to present small windows with additional selection lists appropriate for some options. These additional lists use the CURWORD function so the TAB and ENTER action is consistent throughout the application. The syntax of the DISPLAY function is

\[ \text{CALL DISPLAY(entryname);} \]

Figure 7 Display Window

It is entirely possible that a user might select a data set and immediately exit the application to generate a report. In this case, a single-spaced report, using all the variables in the data set, is generated. When the user customizes the report, choosing options or changing options, the information that the developer needs (to generate the correct report must) be collected and correctly maintained. As an option is selected, the value of a corresponding SAS macro variable is updated.

The macro variable values are available throughout the SAS session. The stored values in the example also determine what the current report illustration should look like. As options are selected or changed, the report illustration is modified. This graphic illustration enables the user to see how the current option request will change the format of the final report. The user may also make changes before the report is actually generated.

When the application is ended, all macro variables are queried. Based on the stored values, the appropriate SAS source code is generated and submitted to the SAS System to create the report. No prior knowledge of the SAS procedure is required.

CONCLUSION

The creation of user-friendly applications requires a flexible programming language that allows for an endless variety of users, tasks, and environments. SCL, available with Version 6 SAS/AF and SAS/FSP software, is a significant step toward helping SAS programmers achieve the appropriate level of interaction, maintain any necessary dialog with users, and interact successfully with other parts of the SAS System.

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