ABSTRACT

Frame entries in SAS/AF use graphic display devices that enable the developer to easily build interfaces which permit users to point and click their way through applications. The functionality of frame entries assists end-users who have limited computer experience to perform activities such as data entry, data processing, and data analysis/reporting.

This paper briefly explains frame entries and describes a data analysis and reporting application called the Consumer Services Information System Menu (CSIS Menu) that was developed with frame entries. The remainder of the paper demonstrates the basic steps used in building the CSIS Menu with a sample application.

INTRODUCTION

SAS/AF frame entries allow developers to build graphic-oriented applications by placing objects such as icons, push buttons and text boxes in a window with the different objects having different functions. Graphic text and text label objects display text which can not be changed by the user. Text entry objects accept user input and display information or program output. Icon and push button objects are used for making selections or executing functions. List box objects display lists of text from which users can make choices. These are only a few of the objects available and application developers may also create their own objects. Frame entries are controlled by associated Screen Control Language (SCL) entries which contain programming called Screen Control Language (SCL).

The Consumer Services Information System Menu (CSIS Menu) is an example of a data reporting and analysis application developed with frame entries. The CSIS Menu was developed to provide the Pennsylvania Public Utility Commission's Bureau of Consumer Services with a user-friendly interface to SAS for their information delivery needs. The CSIS Menu allows users to create frequently requested reports, known as standard reports, and ad-hoc reports such as frequencies and crosstabs analyzing consumer complaint data from SAS data sets. Users can click their way through menus to choose the desired standard reports and then from the window for that report select required parameters such as year and month before submitting the request and printing the output. For ad-hoc reports, users make the appropriate menu selections to choose the desired type of report, such as a crosstab. Then from the report window, the user makes selections from lists of variables and values to select the data to be analyzed, variables to be included in the report and sorting variables. The report is submitted for processing and the output is printed from this window.

The following step-by-step sample application was developed to illustrate the basic steps used in building the CSIS Menu. This example provides a working application designed for developers who have little or no experience with frame entries. Although the techniques used in the example are very basic, they provide a basis for developing a more complex application like the CSIS Menu.

This sample application includes five frame entries and their associated SCL entries. The first frame is the opening menu, which includes the application title and two icons. Selecting the icons will branch the application to a report selection menu or exit the application. The second frame entry is a report selection menu with two icons, which upon selection, branch to report frames from which information for the report is collected and the report is submitted for processing. The report selection menu also has an icon for going back to the main menu.

The two report frames contain four push buttons, one icon and one text entry object. One button is for branching to a frame with a list box from which the year is selected to subset the data for the report. The other three push buttons are for submitting the report for processing, printing the output and clearing the output window. The icon is for going back to the report selection window.

1 My thanks to the Consumer Services Information System Project at Penn State for giving permission to use the CSIS Menu in this paper and presentation.

2 The sample application presented in this paper was developed with SAS Release 6.11 for the Windows operating system.
text entry object displays the year selected for the report. The fifth frame entry in the application contains a list box with values for the variable "year" that users choose to subset the data. This frame also contains a push button that accepts the list "year" selection and returns to the application frame.

This sample application uses a SAS data set named TEST in a SAS library named SASUSER. The data set contains observations with the numeric variables month, qtr and year. Month has the values 1 through 12. QTR has the values 1, 2, 3 and 4. Year has the values 91, 92, 93 and 94. The data set also contains the character variable status which has the values OPEN and CLOSE indicating the status of the complaint. To use the sample application as presented in this paper, a SASUSER.TEST data set must be created prior to submitting the reports for processing.

FRAME ENTRY OBJECT TERMINOLOGY AND LOGIC

Before starting the sample application, a brief review of frame entry terminology and logic may be helpful. For a more in-depth explanation of frame entry terminology and logic, see the listed references.

When building a frame entry, objects are selected from templates known as classes and put in the frame. Classes identify the actions which can be performed with objects. These actions are known as methods. Classes also identify the attributes such as the name, color, size and placement of the object. These attributes are stored in what are called instance variables. Each type of object, such as an icon or push button has its own class, icon class and push button class. These classes are subclasses (or children) of a parent class called the widget class, which is a subclass of the object class. The importance of the class hierarchy is that subclasses inherit the attributes and methods of their parents and may have methods or attributes of their own. Note that objects such as icons and push buttons are often called widgets. A widget being a displayable object. However, in this paper displayable objects will simply be referred to as objects.

BUILDING THE SAMPLE APPLICATION

The first step in building the application is to invoke the AF Build procedure by issuing the BUILD command from the command line. This command will open a window containing the library, catalog and entry lists as seen in Figure 1.

![Figure 1: Build Library, Catalog and Entry list window](image1)

To create a new catalog for the frame and SCL entries in this example, highlight the SASUSER library and from the menu bar choose FILE, NEW and CATALOG. In the Catalog window, enter the name SUGI21 and select OK.

MAIN MENU FRAME

The first frame entry is created by highlighting the SUGI21 catalog and choosing FILE, NEW and ENTRY from the menu bar. In the New Entry window, enter the name MENU, select the entry type FRAME and select OK to open the Build: Display window for building the Menu.Frame entry. When an entry is first created, the Build: Display window only contains a master region bordered with solid or broken lines which encompass the entire window. The objects put in a frame entry are contained within regions which are placed inside the master region. The Build: Display window for the Menu.Frame entry is displayed in Figure 2, as it will look when the objects are placed in the frame.

![Figure 2: Main Menu Build: Display window](image2)

The Menu.Frame entry in this sample application contains two graphic text objects for the title of the application and two icon objects, which upon selection go to a menu of reports or exit the application.
The first graphic text object containing the first line of the title is created by selecting MAKE, GRAPHIC TEXT and OK from the master region pop-up menu. After making these selections a new region outlined with dashed lines will appear in the master region. Using the mouse, place the new region in the upper middle of the screen with a mouse click. After placing the object, a Graphic Text Attributes window will appear as shown in Figure 3.

![Figure 3: Graphic Text Attributes window](image)

The object attributes such as the object name, text and color are entered in this window. For the first graphic text object in this example, enter the following attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Text</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title1</td>
<td>REPORT GENERATION</td>
<td>Foreground</td>
</tr>
</tbody>
</table>

After entering the attributes select OK to exit the Graphic Text Attributes window.

**Note:** If you need to re-open an Object Attributes window select OBJECT ATTRIBUTES from the object region pop-up menu. Open the pop-up menu with a right mouse click in the object's region.

The second graphic text object, containing the second line of the title, is created the same way the first graphic text object was created using the following attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Text</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title2</td>
<td>Sample Application</td>
<td>Foreground</td>
</tr>
</tbody>
</table>

Creating the icon objects for the Menu.Frame entry is very similar to creating the graphic text objects. The only differences are choosing ICON from the selection list instead of graphic text and selecting a specific icon from the attributes window. To create the REPORTS icon select MAKE, ICON and OK from the master region pop-up menu. After placing the icon, enter the following name and label attributes in the Icon Attributes window as shown in Figure 4.

![Figure 4: Icon Attributes window](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Icon Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit</td>
<td>EXIT</td>
<td>111</td>
</tr>
</tbody>
</table>

After entering the name and label attributes and selecting the icon an additional attribute needs to be entered to tell SAS what command to execute when the icon is pushed. Choose COMMAND PROCESSING from the Additional Attributes area of the Icon Attributes window. This will open the Command Processing window where you will type CANCEL in the Execute SAS Commands on Selection area. Select OK to exit the Command Processing window and OK again to exit the Icon Attributes window.

Now that the Menu.Frame entry in this example is created, the SCL entry that controls the frame can be created. Select LOCALS and EDIT SCL SOURCE from the menu bar. In the Build: Source window that opens, enter the following lines of SCL.

```sas
INIT:
return;
MAIN:
return;
REPORTS:
call display('reports.frame');
return;
TERM:
return;
```
Notice that the SCL for the Menu.Frame entry has four sections, INIT, MAIN, REPORTS, and TERM. Each section begins with the section name followed by a colon and ends with a return statement. The INIT, MAIN and TERM sections are the three basic sections which many but not all SCL entries will have. The INIT section contains programming that is executed when the frame entry is initiated and the TERM section contains programming that is executed when the frame entry is terminated. The MAIN section will execute every time the Enter key is pressed or an object in the frame is selected with the mouse. However, these sections do not contain any programming because it is not needed for the Menu.Frame entry of this application. When there is no programming in these sections, they do not need to be included in the entry. The REPORTS section is executed when the REPORTS icon in the frame is selected. This section contains the line call display('reports.frame'); that calls the Reports.Frame entry which will be created next. The REPORTS section of this SCL entry is typical of program sections in SCL entries. It corresponds to an identically named object in a frame entry. When the object is selected the section of SCL with the same name is executed.

After typing in the SCL, compile the program by selecting LOCALS and COMPILe from the menu bar. After the compile is completed, a message will appear at the bottom of the Build: Source window reporting the results of the compile. If there are errors they will be recorded in the Message window. When the SCL is successfully compiled, exit the Build: Source and Build: Display windows and return to the main Build window.

REPORTS FRAME

Now that the frame entry and corresponding SCL entry for the main menu of the sample application are created, the entries for the reports frame need to be created. The Reports.Frame entry is created the same way as the Menu.Frame entry. With the SUGI21 catalog highlighted, choose FILE, NEW and ENTRY from the menu bar. In the New Entry window enter the name REPORTS, select the entry type FRAME and select OK. The empty Build: Display window for the Reports.Frame will appear. This frame will be another menu containing three icons labeled STATUS BY MONTH, STATUS BY QTR and GO BACK as shown in Figure 5.

![Figure 5: Reports Menu Build: Display window](image)

The first two icons are for choosing one of two reports and the GO BACK icon is for returning to the main menu. As before, the icon objects are created by choosing MAKE, ICON and OK from the master region pop-up menu. Use the following attributes for the REPORTS and GO BACK icons.

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Icon #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month</td>
<td>STATUS BY MONTH</td>
<td>405</td>
</tr>
<tr>
<td>Qtr</td>
<td>STATUS BY QTR</td>
<td>405</td>
</tr>
<tr>
<td>Goback</td>
<td>GO BACK</td>
<td>116</td>
</tr>
</tbody>
</table>

Like the EXIT icon in the main menu, the GO BACK icon needs additional information entered in the attributes window to tell SAS what command to execute when the icon is pushed. As before, choose COMMAND PROCESSING from the Additional Attributes area, then type CANCEL in the Execute SAS Commands on Selection area of the Command Processing window. Select OK to exit the Command Processing window and OK again to exit the Icon Attributes window.

After the Reports.Frame entry is finished the SCL entry for the frame must be created. From the menu bar select LOCALS and EDIT SCL SOURCE to open the Build: Source window for the Reports.Scl entry and enter the following programming.

```scl
MONTH:
call display('statmont.frame');
return;

QTR:
call display('statqtr.frame');
return;
```

The Reports.Scl entry contains two sections, MONTH and QTR. Each section calls a frame from which a report is produced when the corresponding icon in the Reports.Frame entry is selected. Notice that there is no INIT, TERM, or MAIN section because there is no need for them. After entering the programming, compile it by
selecting **LOCALS** and **COMPILE** from the menu bar. Then exit the Build: Source and Build: Display windows and return to the main Build window.

**STATUS BY MONTH REPORT FRAME**

With the main menu and reports menu built it is time to create the frame from which the Status by Month report is submitted. With the SUGI21 catalog highlighted, choose **FILE**, **NEW** and **ENTRY** from the menu bar. In the New Entry window enter the name **STATMONT**, select the entry type **FRAME**, and select **OK**. The empty Build: Display window for the Statmont Frame will appear. When complete this frame will contain four push button objects, one text entry object and one icon object as shown in Figure 6. The push button labeled YEAR is for calling a frame containing a list box for choosing the year of the report while the text entry object displays the year selected. Prior to the selection of a year the text entry box displays "REQUIRED" indicating a year selection is required. The buttons labeled RUN, PRINT and CLEAR OUTPUT do just what their labels indicate. The GO BACK icon returns to the previous frame.

![Figure 6: Status by Month Report Build: Display window](image)

Figure 6: Status by Month Report Build: Display window

Objects in this frame are created the same way objects were created in the other frames. Select **MAKE**, the type of object (**PUSH BUTTON**, **ICON**, or **TEXT ENTRY**) and **OK** from the master regions pop-up menu. Use the following attributes for the push button objects.

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>YEAR</td>
<td>Char</td>
</tr>
<tr>
<td>Run</td>
<td>RUN</td>
<td>Char</td>
</tr>
<tr>
<td>Print</td>
<td>PRINT</td>
<td>Char</td>
</tr>
<tr>
<td>Clear</td>
<td>CLEAR OUTPUT</td>
<td>Char</td>
</tr>
</tbody>
</table>

Use the following attributes for the icon object.

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goback</td>
<td>GO BACK</td>
<td>Char</td>
</tr>
</tbody>
</table>

Like the GO BACK icon in the Reports Frame, additional information needs to be entered in the attributes window to tell SAS what command to execute when the icon is selected. As before, choose **COMMAND PROCESSING** from the Additional Attributes area, then type **CANCEL** in the Execute SAS Commands on Selection area of the Command Processing window. Select **OK** to exit the Command Processing window and **OK** again to exit the Icon Attributes window.

When creating the text entry object the Text Entry Attributes window will appear as shown in Figure 7.

![Figure 7: Text Entry Attributes window](image)

Figure 7: Text Entry Attributes window

Enter **Y** as the **Name** attribute for this text entry object. Now an initial value needs to be assigned to the object. From the Additional Attributes area of the window select **INITIAL VALUES/FORMATS**. In the Initial Values/Formats window as shown in Figure 8, enter **REQUIRED** in the Value area of the window and select **CHARACTER** in the Data Type area of the window. Select **OK** to exit the Initial Values/Formats window and **OK** again to exit the Icon Attributes window.

![Figure 8: Initial Value/Formats window](image)

Figure 8: Initial Value/Formats window

As with the other frame entries in this application, after the objects are made the SCL needs to be entered in the SCL entry for the frame. Open the Build: Source window by selecting **LOCALS** and **EDIT SCL SOURCE** from the menu bar and enter the following SCL.
YEAR:
call display('yearlist.frame',y);
return;

MAIN:
if run='RUN' then do;
if y='REQUIRED' then do;
   _msg_='NOTE: You must select a year before
running the report.';
   return;
end;
   _msg_='NOTE: Please wait while your request is
processed.';
   refresh;
   submit continue;
end;
return;

PRINT:
rc=woutput ('print',sasuser.profile.default');
rc=woutput ('clear');
return;

CLEAR:
rc=woutput ('clear');
return;

TERM:
rc=woutput('clear');
return;

The Statmont.Scl entry has five sections, including YEAR, MAIN, PRINT, CLEAR and TERM. The YEAR section contains a line that calls the entry Yearlist.Frame and passes the variable Y containing the value of year used in the title and the SET statement of the report.

As mentioned before, the MAIN section will execute any time the Enter key is pressed or an object in the frame is selected with the mouse. The MAIN section has four functions. First, it verifies the RUN button was selected before submitting the report for processing. Second, it then checks to see if a year is selected and warns the user if not. Third, the program sends the user a message indicating the report is being processed. Finally, the report is submitted for processing.

The PRINT section prints the contents of the output window and then clears the output window. The CLEAR section clears the contents of the output window and the TERM section clears the output window when the frame is terminated.

After the programming is entered, compile the program by selecting LOCALS and COMPIL from the menu bar. Then exit the Build: Source and Build: Display windows to get back to the main Build window.

STATUS BY QUARTER REPORT FRAME

After creating the Statmont.Frame and Statmont.Scl entries for the Status by Month Report they can easily be copied and edited to create the Statqtr.Frame and Statqtr.Scl entries for the Status by Quarter report. This process includes copying the entries, changing the description of the new entries and making changes to the SCL programming in the SCL entry. Highlight the Statmont.Frame and Statmont.Scl entries and select the FILE and COPY commands from the menu bar to display the Copy window as shown in Figure 9.

Tab down to the New Name area and type in STATQTR for both the frame and SCL entries and then select OK. Notice that the descriptions for the new entries were not entered during the copy process. After the copy is completed the new entries have the same descriptions as the original entries. The descriptions will be changed next with rename. After the copy window closes, highlight the new entries Statqtr.Frame and Statqtr.Scl, and select FILE and RENAME from the menu bar to display the rename window as shown in Figure 10.

Figure 9: Copy window

Figure 10: Rename window
To change the description of the entry in the Rename window, change STATMONT to STATQTR for both the frame and SCL entries and click on OK.

Now that the Statqtr.Frame and Statqtr.Scl entries have been created, the necessary modifications to the SCL entry can be made. Open the Build: Display window for the Statqtr.Frame entry by highlighting the entry and selecting FILE, OPEN and EDIT from the menu bar. From the menu bar select LOCALS and EDIT SCL SOURCE to open the Build: Source window. Listed below is the SCL programming with the modifications in bold.

```sas
YEAR:
call display('yearlist.frame.y');
return;

MAIN:
if run='RUN' then do;
  if y='REQUIRED' then do;
    _ffiSS_='NOTE: You must select a year before running the report.';
    return;
  end;
  _ffiSS_='NOTE: Please wait while your request is processed.';
  refresh;
  submit continue;
end;

options pageno=1;
title! 'STATUS BY QUARTER FOR YEAR 19&y';
data one;
set sasuser.test (keep=year qtr status where=(year=&y));
run;
proc freq data=one;
tables status * qtr;
run;
endsubmit;
end;
return;

PRINT:
rc=woutput ('print', 'sasuser.profile.default');
rc=woutput ('clear');
return;

CLEAR:
rc=woutput ('clear');
return;

TERM:
rc=woutput ('clear');
return;
```

After making the modifications compile the programming by selecting LOCALS and COMPILE from the menu bar and then exit the Build: Source and Build: Display windows.

YEAR SELECTION FRAME

The last frame entry is the Yearlist.Frame entry for selecting the desired value of year to subset the data for the report. With the SUGI21 catalog highlighted choose FILE, NEW and ENTRY from the menu bar. In the New Entry window enter the name YEARLIST, select the type FRAME, and select OK. The empty Build: Display window for the Yearlist.Frame will appear. When completed this frame will consist of a list box containing values for the variable year and a push button labeled OK as seen in Figure 11.

![Figure 11: Yearlist Build: Display window](image)

Select MAKE, LIST BOX and OK from the master regions pop-up menu. After placing the object, the List Box Attributes window will appear as shown in Figure 12.

![Figure 12: List Box Attributes window](image)

In the Attributes window enter the following attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>YEAR</td>
</tr>
</tbody>
</table>

Entering the values in the list box is done by clicking on the down arrow to the right of Enter values... in the List Box Population area of the List Box Attributes window. Then select ENTER VALUES... to display the Enter Values window as shown is Figure 13.
When the Enter Values window appears for the first time it contains items entered when the list box was created. These items can be edited or deleted. For this example they will be deleted before adding the desired items. Select the Actions button and Delete All from the menu. This will display the Delete Verification window where you select OK to delete all the highlighted items. To add new items select the Actions button and Add mode on from the menu to turn on the add mode.

Now enter the first item by typing 91 in the Item area of the window where the cursor is located and click on the NEW button. The item will be placed in the List Box Items area of the window. The other items - 92, 93 and 94 - are added the same way. After all items are entered click on OK to exit the Enter Values window and OK again to exit the List Box Attributes window.

Creating the OK push button is done as before, by choosing MAKE, PUSH BUTTON and OK from the master region pop-up menu. Like the EXIT and GO BACK icons in the other frames, this OK push button needs command processing information telling SAS what command to execute when the push button is selected. As before, choose COMMAND PROCESSING from the Additional Attributes area, then type END in the Execute SAS Commands on Selection area of the Command Processing window. Select OK to exit the Command Processing window. After entering the following attributes for the push button select OK to exit the Icon Attributes window.

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ok</td>
<td>OK</td>
</tr>
</tbody>
</table>

After completing the Yearlist.Frame entry, create the Yearlist.ScI entry by selecting LOCALS and EDIT SCL SOURCE from the menu bar. Enter the following programming in the Build: Source window and compile it by selecting LOCALS and COMPIL3 from the menu bar.

```sas
ENTRY Y$;
YEAR:
call notify('year', _get_last_sel', x, issel, y);
return;
```

The Yearlist.ScI entry contains the line ENTRY Y$; and the YEAR section. The first line is necessary for passing the variable Y between the Yearlist.Frame and the previous frame, Statmont.Frame or Statqtr.Frame. The YEAR section uses a list box method to retrieve the text from the selected row of the list box object.

After the code is compiled, exit the Build: Source and the Build: Display windows to go back to the main Build window. From here the application can be tested by highlighting the Menu.Frame entry and selecting LOCALS and TESTAF from the menu bar. The Testaf procedure is similar to running the application, however the reports will not be submitted for processing. The Testaf procedure will record any errors in the application in the Message window.

When the application building and testing is completed close the Build window and return to display manager. The application may be invoked entering AF C=SASUSER.SUGI21,MENU.FRAME in the command line. As mentioned earlier, to use the sample application as presented in this paper, the SASUSER.TEST data set must exist prior to submitting the application's reports for processing.

**SUMMARY**

This step-by-step sample application illustrates the basic steps used in building a data reporting and analysis application with SAS/AF frame entries. The application could be described as an interface that collects user input to modify SAS programming that is submitted for processing. While the techniques used in the application are very basic, they provide the basis for developing a more complex application. For more in-depth information on SAS/AF frame entries and Screen Control Language, see the following references.
REFERENCES


ACKNOWLEDGEMENTS

I would like to thank my colleague, Tura Eisele, for reviewing this paper and offering several useful suggestions. Also thanks to my wife, Patty, for help with editing.

SAS and SAS/AF are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks and trademarks of their respective companies.

The author may be contacted at:

The Pennsylvania State University
314 Shields Building
University Park, PA 16201
Phone: (814) 863-0700
Internet: VLT@PSU.EDU