**ABSTRACT**

The SAS(R) System for the Microsoft Windows(R) environment has the capability of embedding Windows objects such as images, sound objects and video clips as well as SAS/GRAPH objects within a SAS/AF FRAME. This tutorial describes how to use these objects to build a SAS/AF application and how to build a graphics drill-down application using available Screen Control Language(SCL) routines. The tutorial is targeted for application builders who wish to enhance user friendly front-ends on their systems.

**INTRODUCTION**

A number of Windows applications allow Object Linking and Embedding(OLE). One example of this is a simple clipboard copy from one application to another. A SAS/AF FRAME can also use OLE such as bitmap images, sounds and video from external Windows applications. In addition, a SAS/AF application can 'animate' a SAS/GRAPH through the use of SCL.

Many of the techniques presented in this paper are tied to the operating environment, Microsoft Windows. This is adequate for building very user friendly front-ends. You should also consider the portability of the application. There is a definite advantage in using SAS add-ons such as SAS/GRAPH, SAS/EIS and SAS/VIDEO for portability to other operating systems.

**STATIC IMAGES**

Release 6.10 can import static images (.TIF, .GIF, .BMP files) directly through SAS/GRAPH(1). This improves the portability of the applications built but it requires that the user licenses SAS/GRAPH. You can use also a Windows application, such as Paintbrush to design graphic text or picture and paste it into SAS/AF FRAME.

To get started, use the Microsoft Paintbrush program to read a .BMP file. When displayed, clip the area of the image and use the EDIT..COPY pull down menu to copy the graphic object to the clipboard.

You can start the SAS/AF build procedure to create a FRAME application:

```
PROC BUILD c=SUG.OLE.PIC.FRAME;
```

You can also start the BUILD procedure from the command line:

```
BUILD SUG.OLE.PIC.FRAME
```

When the FRAME is displayed, choose ACTIONS..MAKE pull down or pop-up menu, then OLE - Paste Special.

The next screen displayed will show a choice for OLE paste. A simple Object will work. Other Data Types can be used and these are documented in the SAS Companion Guide for the Microsoft Windows Environment(2).

Once you have selected Paste, the OLE Paste Special Attributes screen will be displayed. The entry name must be specified as any valid SAS name. In this case FACE:
Press on OK and the picture is on the SAS/AF FRAME. This object is called FACE. The source SCL code, accessed through the LOCALS...EDIT SCL source pull down menu (or pop-up) can be used when this object is clicked on.

You can also program which object appears on the frame with SCL provided that several OLE objects are defined. For example, if two objects, 'FACE1' and 'FACE2' are defined OLE objects, they can be located on the same area of the frame, and SCL can select which is displayed. The following SCL shows how a radio button called CHOICE with default values 'Button1' and 'Button2' is used to selectively display either 'FACE1' or 'FACE2':

INIT:
* do not display until called ;
call notify('face1', '_hide_');
call notify('face2', '_hide_');
RETURN;

CHOICE:
call notify('face1', '_hide_');
call notify('face2', '_hide_');
if choice='Button1' then call notify('face1', '_unhide_');
if choice='Button2' then call notify('face2', '_unhide_');
RETURN;

SOUND OBJECTS

An excellent article dealing with sound objects appeared in "Observations" magazine(3). Some of the applications include verbal help, warnings and user friendly welcome frames. In this example, a warning message 'Remote Has Disconnected' will be used when a SIGNOFF command is encountered. Sound can only be played through your PC if the appropriate driver is installed. For those without a sound card installed, a driver is (Speaker.drv) is available from Microsoft that will play back sounds at a lower quality through the PC speaker.

Creating a sound clip in a SAS/AF FRAME involves the creating of an OLE object with a sound application, such as the Sound Recorder. In this example, the RMTDISC.WAV file contains the sounds 'Remote has disconnected'. The RECORDER application EDIT pull down menu must be used to COPY to the object. This create the OLE object that SAS/AF can use.

Start SAS/AF with a BUILD command:

BUILD SUGI.OLE.SOUND.FRAME

When the SAS/AF FRAME is displayed, choose MAKE from the ACTIONS pull down or pop-up menu, then choose OLE - Paste Special. Once you have selected PASTE button, the OLE paste special attributes window opens.

The name of the object is changed from OBJxx default to WARNING. You must also fill in the name of the entry, in this case, WARNING. The default SAS/AF Frame will display the Sound Recorder icon.
This object can be hidden with SCL, and called when required. For the purpose of this tutorial, the sound clip WARNING is played when the SIGNOFF command is issued during a SAS/CONNECT session. The following SCL was used:

```
INIT:
   control enter;
   call notify("warning", "_hide_");
RETURN;

MAIN:
   if word(1, 'U') = 'SIGNOFF' then
      call notify("warning", "_execute_", "play");
RETURN;
```

**VIDEO CLIPS**

Windows provides a Media player application in the ACCESSORIES group. The Media Player requires that a video driver be available when playing video clips. A video driver such as Video for Windows (VFW) can be used with standard VGA screens. To bring a video clip into SAS/AF Frame, start the media player and open a video file (WINDSURF.AVI).

Options in the Media player should be set to PLAY in CLIENT Document. Use the EDIT, COPY Object in Media player to create the OLE object, then start SAS/AF with a BUILD command:

```
BUILD SUGI. OLE VIDE0.FRAME
```

When the FRAME is displayed, chose MAKE from the ACTIONS pull down or pop-up menu, then chose OLE - Paste Special.

The next dialog box requires that a name be supplied to the HSERVICE, for example WINDSURF. The object name can also be changed from it default OBJxx to WINDSURF. The video clip is now part of the SAS/AF application.

The video clip can be started with SCL using CALL NOTIFY:

```
CALL NOTIFY("WINDSURF", "_EXECUTE_", "Play");
```

The video clip can be used as a welcome screen, by executing the object in the INIT section of the SCL. The INIT section of the SCL, which is executed prior to the frame being displayed results in the video object being played outside of the frame. To get around this, a _POSTINIT_ method defined for the parent Widget class can be used on the video OLE object. This is documented in the SAS/AF Frame Entry Reference(4)

```
INIT:
   call notify("windsurf", _postInit_);
   call notify("windsurf", _execute_", "play");
RETURN;
```

It is not possible to make SAS aware of when the OLE Video object is done executing. Clicking on another object on the same frame while a Video clip is playing can cause general protection faults. You can prevent this by building a delay after the execute stage, with the SCL WAIT routine:

```
   call notify("windsurf", _execute_", "play");
   call wait(5); 
```

The wait time interval is estimated from the running time displayed on the Media Player.

Several video clips can be put on the same area of the FRAME (Overlay) and SCL can be used to control which video clip is played. In one application three video clips were added to a FRAME, object names VIDEO1, VIDEO2 with corresponding ENTRY names. Another object, a RADIO button called BUTTON controlled which video would be displayed. The SCL includes WAIT routines and _UPDATE_ methods to the HSERVICE entries:

```
INIT:
   call notify("video1", _hide_);
   call notify("video2", _hide_);
RETURN;

BUTTON:
   if BUTTON="VIDEO1" THEN DO;
      call notify("VIDEO3", _update_, "SUGI. OLE VIDEO1.HSERVICE");
      call notify("VIDEO3", _execute_", "play");
```

```
BUILD: DISPLAY VIDEO.FRAME (E) 
```
SAS/GRAPH ANIMATION

SAS/Graph objects offer a 'natural' connection with SAS/AF and applications built purely in SAS tend to be highly portable across operating systems. For the purpose of this tutorial, a simple SAS/GRAPH X-Y plot will be embedded into a SAS/AF FRAME and animation in the form of a moving line plot will be done.

Start SAS/AF with a BUILD command:

BUILD SUGI.GRAPHPLOT.FRAME

When the FRAME is displayed, chose MAKE from the ACTIONS pull down or pop-up menu, then chose GRAPHICS. You will need to fill the Graphics Attribute Screen with the following information:

- NAME:
- PLOT TYPE: JOIN
- DATA SET: SUGI.PLOT
- X VARIABLE: TIME
- Y VARIABLE: PERCENT

The SUGI.PLOT data set contains 100 observations of random PERCENT for TIME values of 1 to 100.

Press OK on the Attribute Screen and the FRAME will then display the SAS/GRAPH plot:

The SCL behind this application uses a simple data set which is updated internally. Normally, this type of data set would be updated externally, through a View for example. The INIT section contains a VARSTAT function that gets the maximum value on the X axis (time). This is used as the starting point to add more observations to the data set. The VARSTAT function is documented in the SCL reference(5). The action is obtained by looping over the update of the data set and updating the frame object inside the loop. The update of the FRAME is controlled by the MOD function. This simulates animation, depending on machine speed.

START:

call wait(12);
END;

IF BUTTON=VIDEO2 THEN do;
   call notify('VIDEO3', 'update',);
   call notify('VIDEO3', 'execute', 'play');
   call wait(12);
END;

RETURN;

SAS/GRAPH DRILL DOWN

SAS/GRAPH objects can be queried in a SAS/AF FRAME application. In this example, a Bar Graph created from SAS/AF Graphics class can be queried by clicking on any bar. The action taken is then to start a FSVIEW session to display the data. This example is based on a similar program developed in the SAS/AF Frames course available from SAS Institute(6).

To begin the process, start SAS/AF with a BUILD command:

BUILD SUGI.GRAPH.DRILL.FRAME

When the FRAME is displayed, chose MAKE from the ACTIONS pull down or pop-up menu, then GRAPHICS. You will need to fill the Graphics Attribute Screen with the following information:

- NAME:
- PLOT TYPE: VERTICAL BAR
- DATA SET: SASUSERHOUSES
- MIDPOINT VARIABLE: STYLE
- RESPONSE VARIABLE: PRICE
- STATISTICS TYPE: MEAN

SAS/GRAPH will generate a vertical bar chart which is displayed on the FRAME.
The graph can be queried using SCL. In this example, when a particular bar is clicked on, an FSVIEW window displays all observations that have the style associated with the vertical bar.

```sas
length dname $57 indepvar $8 indtype $1;
GRAPH:
call notify('graph', '_get_value_\'valueid\');
call notify('graph', '_get_dsname_\'dname\');
call notify('graph', '_get_indep_var_\'indepvar\');
indtype=getitem(valueid,'indtype');
if indtype='c' then do;
   indval=getitem(valueid,'indvalue');
   w=(where="\" indepvar \"=" \" \" indval \" \" \" \")
   _msg = w;put _msg;
   end;
else do;
   indval=getitemn(valueid,'indvalue');
   w=(where="\" indepvar \"=" \" \" indval \" \" \")
   _msg = w;put _msg;
   end;
call wrviewport(8,33,22,75);
dname=compress(dname || w);
   _msg = dname;put _msg;
call fsview(dname);
return;
```

When this SAS/AF program is executed, the bar graph is displayed. When a specific bar is clicked, an FSVIEW window is displayed.

SAS/EIS software can also be used in developing drill down applications, without the SCL programming. This saves a considerable amount of time in development, but unlike a SAS/AF FRAME application, it requires that SAS/EIS be licensed at each user workstation.

CONCLUSION

OLE objects offer exciting enhancements in the building of SAS/AF applications for SAS under operating in the Microsoft Windows Environment. These objects include images, sounds and video. SAS/GRAPH objects offers more portability, and the advantage of several SCL routines that can be used to animate graphs and build drill down applications.

REFERENCES

3. Using OLE with SAS/AF FRAME Applications in the Microsoft Windows Environment (Gilmore, Wagner) Observations, Fourth Quarter 1993
5. SAS Screen Control Language: Reference, Version 6, First Edition
6. Building SCL Applications Using FRAME Entries Course Notes

ACKNOWLEDGMENTS

The author wishes to thank John Kirtz, of Snowy Owl Software and Loretta Golby of ISM Alberta for their many constructive comments in the preparation of this paper.

Please address comments or questions to:

Serge Dupuis
BKD SOFTWARE CONSULTANTS
INTERNET: 75032,2240@COMPUTER.COM

SAS, SAS/AF, SAS/FSP, SAS/VIDEO, SAS/CONNECT, SAS/GRAPH, SAS/EIS are registered trademarks of SAS Institute Inc. in the USA and other countries. (R) indicates USA registration.

Other brand names are registered trademark or trademarks of their respective companies.