ABSTRACT

With the introduction of the SAS/AF software FRAME entry, developers are now able to incorporate graphical user interface (GUI) and object-oriented programming (OOP) methodologies in the development of applications. Objects can now be drawn from the developer's toolkit and used to develop GUI-based front ends. The SAS/AF software's list box and push-button facilities enable users to interact with applications much more efficiently and intuitively. Not only is this technology powerful, but it is fun and exciting as well!

INTRODUCTION

The Biostatistics Study Management System is a prototype system that was created by the author in order to become familiar with the SAS/AF software FRAME entry used in conjunction with Screen Control Language (SCL). Playing with high-tech 'toys' prior to developing 'live' applications is her modus operandi. This paper's primary purpose is to not only garner interest in this exciting technology, but to also demonstrate a practical use for the product. Nonetheless, the author strongly recommends that you take the SAS/AF software FRAME entry course to capture some of the necessary basics of the product that will not be discussed here. Another good source of information is a 4 part tutorial series given by Jeff Cartier, et al at SUGI 18.

SYSTEM DEVELOPMENT

The first step in developing the prototype was to create a few 'CLASS' objects that could be created once but used many times. This concept is similar to creating a library of macros that can be 'called' upon demand.

The first two 'CLASS' objects created were the Graphics Text 'Biostatistics Study Management System' logo and the 'SAS Programming Group' logo. Once this step was completed, the author saved these objects as resource entries in the resource list. A developer can then access these logos using the 'pop-up' resource list that appears when a 'fill' request is made. Refer to the SAS/AF Software: Frame Entry Usage and Reference manual, chapter 10 and Part III of Jeff Cartier's paper for more details.

This is an excellent way to build permanent 'reusable' objects, such as push-buttons, icons, etc.

The Main Menu was developed by accessing the two logo 'CLASS' objects created above from the resource list. The next step was to create 6 regions or 'objects' to represent the 'MAIN MENU' graphics text object and the 5 main menu items. Figure 1 illustrates the prototype Main Menu. The SAS® software supplied icons were used to fill the regions created.

SCL programs were then associated with this menu and with the 5 main menu objects on this GUI-based window. You may notice in the SCL code for the Main Menu in the appendix that there is not any code associated with the EXIT icon label. The author chose to demonstrate an easy way for a developer to implement the execution of a SAS command when the user clicks upon an object. This was done by using the LOCALS pulldown menu to access the OBJECT ATTRIBUTES display. The developer then clicks on COMMAND...
PROCESSING and simply type END as the SAS command to be executed when the object is selected.

Another alternative would be to create a 'CLASS' object as discussed earlier. Since the exit function is rather common, this can be stored as a 'reusable' object.

Due to the dynamic nature of the BSMS database, the author opted to fill or populate the INDICATION and THERAPY list boxes by pointing to the Indication(INDIC) and Therapy(THRPY) datasets via the List Box Attributes screen. The INDIC and THRPY SCL variables contain the id of the opened datasets as defined in NEWSTUDY.SCL. This will ensure that the list boxes will be populated with the most current version of these datasets.

When the user clicks on the 'ADD' button, the VAX/VMS directories are defined and set up for the newly-created study. One can simply adapt this section of the SCL code to the operating system that is in use at his or her installation.

In addition to the creation of study directories, the application's PROTOCOL and DRUG tables in the database are updated immediately. The drawbacks are obvious and a couple of necessary enhancements would be to allow an audit trail, the ability to 'commit' and 'rollback', and other essential 'database management' functions. Whether to use the SAS software or a DBMS is outside the scope of this discussion but it is something every system person should consider in conjunction with their user community.

The 'CANCEL' button allows the user to back out of the system without saving any of the modifications made to the screen.

Finally, if the user clicks on the 'HELP' button, Figure 4 is displayed. The contents of this screen were created...
using the HELP entry.

CONCLUSION

The ease with which this prototype system was developed has impressed the author tremendously. Being able to actually use the concepts of OOP and GUI to develop a product can be very fun and exciting. The author's intent is to invest more time in the analysis and design phase in order to fully develop user specifications, normalize the database, add many essential features, and integrate the system with a few of the current subsystems used at Syntex. This system will go from being a working rapid prototype model to being a complete working system with reusable objects.

ACKNOWLEDGMENTS

I wish to thank Ms. Margaret Schrempf for her feedback in developing this prototype. I would like to especially thank SAS Institute Inc. for enhancing the SAS/AF product with the Frame Entry.

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

REFERENCES


SCREEN CONTROL LANGUAGE (SCL)

SOURCE CODE

ENTRY: MAIN.SCL  SCL for Main Menu

```
ENTRY: MAIN.SCL  SCL for Main Menu
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```

ENTRY: MAIN.SCL  MAIN STUDY

```
ENTRY: MAIN.SCL  MAIN STUDY
```

ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL

```
ENTRY: STADM.SCL  SYSTEM ADMINISTRATION SCL
```
/* Determine position number of the DRUG */
/* variable in DRUGS dataset then look for */
/* DRUG entered by user. */

vdrug=varnum(drugs,'Drug');
valid=locate(drugs,vdrug,drug);
return;

TRADQAH:
return;

PRODESC:
return;

COMMENT:
return;

THERAPY: /* Execute when a therapy is selected */

/* To store the currently selected row's */
/* text in the THERAPY variable */
/* call notify('therapy',get_last_sel_row,thapy) */
vther=varnum(protds,'therapy');
return;

INDIC: /* Execute when an indication is selected */

/* call notify('indic',get_last_sel_row,indic) */
vidic=varnum(protds,'indic');
return;