SAS/FSP Software, SAS/AF Software and SAS/DMI Software
as tools for work stations

David T. Bean
Morino Associates, Inc.
Vienna, VA

The Institute Program Products (IPPs) have become some of the most popular tools for use by Information Center personnel during the past decade. With the introduction of the menuing facilities and data entry facilities of SAS/FSP, SAS/AF and most recently SAS/DMI, even more application areas have been opened for the use of the IPPs.

SAS/FSP provides basic data browse and edit capability into data sets. User defined screens can be used to provide a customized layout to the data. SAS/AF was designed to be an Information Center tool to provide basic menuing capability. Simple decision logic has been built into the product allowing the application user to select applications or options to run.

SAS/DMI was originally written to interface IBM's Interactive System Productivity Facility (ISPF) tables with SAS data sets. The powerful nature of the menuing, selection and display capabilities of ISPF has made SAS/DMI strong candidate to Information Center applications as well. User defined validation rules and the interface between the SAS data step and the panel display facility of ISPF can provide powerful cross field validation capability.

Introduction

Information Center managers are constantly faced with application users who possess a wide range of unique application specific requirements. Choosing the appropriate type of software is usually significant proportion time spent in the solution of the application solution planning. This paper will use each of the above to illustrate how the selection of the appropriate package will improve the overall acceptance of the application.

To motivate the discussion, Consider the plight of the SAS Representative in a large Information Center. Users are developing applications, others are using applications which have been implemented by IC personnel, and still others are using SAS as an adhoc reporting tool. With all of these users there is always questions about:

"I've had this type of problem. Is it a bug or am I doing something wrong?"

The Institute provides a early warning vehicle to assist in the answering of this type of question. SAS Usage Notes provide the administrator with a data set about most the know problems in the program code and user documentation which the Institute supports. Our goal here is to develop an inquiry system which will permit the SAS administrator to query the notes.

Hypothetical Problem

The requirements for this hypothetical system are simply to provide a full screen interactive environment which permits the easy access to the Usage Notes. In addition the administrator must be allowed to search the data for observations (records) of interest. Furthermore the application user would like to view ZAP data associated with any usage notes, if one exists. ZAPS are actual program code changes which the Institute will make available to users if possible. ZAPS may or may not exist for a known program problem.

Three possible solutions the above requirements come to mind almost immediately.

Solution 1 - Use the SAS/FSP procedure FSBROWSE along with customized screens to provide an adhoc vehicle to "search" the data.

Solution 2 - Use SAS/AF to provide menuing facility to permit the selection of data, browsing of notes and support the selection ZAP data.

Solution 3 - Use SAS/DMI as an integrated solution. A solution which will provide a means to scan the data, apply multiple keys to serve as a criteria for selection and viewing. In addition permit the direct browse of the ZAP data.
Solution 1

Using the FSBROWSE procedure, the first solution is one in which there is very rapid program development. In fact the Institute provides an FSBROWSE screen for this solution which has already been customized to permit the easy viewing of the notes. Figure 1 demonstrates the screen which is currently provided. We will use this as our first solution.

There are three major "features" in this solution which are the following:

- KEYS which can be searched using the FSBROWSE "S" command
- Context keys which can be used to "locate" only the observations of interest. (e.g. Locate problems with SAS Macros)
- Application development is minimal, even if you choose not to use the Institute provided screens. The procedure has a self contained screen design dialog

Attributes of this applications search capability are limited only to the extent that one must "Search" or "Locate" one one variable at a time. For instance, the application user must "search" through every observation which has the same keys. One might conceive however, of an application user who was interested in searching on the key SYS.MACRO and the key of ABENDOC4 at the same time.

Using the SAS/FSP product by itself, however, we find that one of the requirements of this problem has not been met. There is no current option within the FSBROWSE to view an external data set as would be required to view any zaps in the ZAPS library. The only available solution is to have the application user "remember" the ZAP member name, exit FSBROWSE, and use PROC FSLIST or ISPF Browse to view the zap data.

The above represents a very quick to develop, but may not be a very acceptable solution to the proposed problem by the application user. A better approach would be to use a menuing facility to select the required information, browse and or print the data. SAS/AF could easily be used to assist in this solution.

Solution 2

SAS/AF collects and neatly stores a series of programs, panels and menu's within a catalog to form an interactive dialog. Dialogs are executed by using the DISPLAY procedure. Hence an immediate benefit to the SAS programmer is the ability to store in one location all the information about a given dialog. The obvious benefit is the simplicity of any ongoing maintenance.
The catalog is developed by using the BUILD procedure. Although there are several member types in the catalog, for this discussion we will only cover the basic MENU, and PROGRAM members. In a wide variety of applications where SAS/AF can be used, the MENU is the controlling mechanism over all subordinate iterations. Hence the most common form of application is on which has a MENU calling a PROGRAM or another MENU. Figure SUGI-007 is an example of a selection menu which presents the application user with three options.

Option 1 - Displays a panel where the user can select and view the entire Usage Note data set using FSBROWSE. Hence the entire solution is simply one leg of the SAS/AF dialog.

Option 2 - Displays another panel which is used to subset the Usage Note data set.

Option 3 - Could use the FSLIST procedure to view the ZAP library.

Figure SUGI-008 demonstrates the attributes panel associated with the menu. Attribute panels are always associated with MENU panels to define the mapping between listed options and the subordinate PROGRAMS or MENUS. In effect a PF key or number is the link to start up PROGRAM or another MENU member.

Figures SUGI-011 and SUGI-012 demonstrates the PROGRAM member USUBSET. PROGRAM members are generally both a panel for data entry and actual program code. In this example, Figure SUGI-011 contains a "fill in the blanks" display in which the application user will be prompted for input. The PROGRAM code in Figure SUGI-012 demonstrates sample coding to conditionally include subsetting IF statements if the user enters data into the corresponding fields. The application user would presumably "fill in the blanks" to obtain a listing of the Usage Note observations that were desired.

This solution turns out to be an automation of the first solution. It would provide the application user with additional value due to the increased function. The usability features were added to provide the application user with better access to the data.

For all the benefit to the user in this application by the search dialog, alas we still have not met all the application requirements. There is still no direct link to the ZAPS library. In addition resource
requirements could become quite high. One could easily imagine, that after searching the data using dialog option 2, that the inquiry may lead to additional inquiries about the same or similar topics. The application user would have to visit dialog option 2 several times, "fill in the blanks", and repass the entire Usage Note data to obtain different views.

Solution 3

The third solution provides the application developer with a considerably wider range of tools to address this type of problem. SAS/DM provides the application developer with a very robust library of CALL functions to the ISPF dialog library. In fact over 63 new functions are available under SAS/DM applications.

Among the available resources are functions to browse or edit data, library manipulation functions, display services and table manipulation services. SAS/DM also provides an access method to IBM's GDDM program product for graphics output.

The final solution to the proposed problem will make use of the ISPF table services available to the application developer. The solution would involve two steps including a one time program to convert the Usage Note data from SAS data set form to that of an ISPF table. Once the Usage Notes are into an ISPF table a simple table display dialog will be written to browse the Usage Notes.

ISPF tables are in concept very similar to SAS data sets. SAS observations translate to rows in a table while SAS variables translate to ISPF column variables. The there are four distinct differences between SAS data sets and ISPF tables. First the ISPF tables reside in a partitioned data set (under the MVS operating system). Second, whenever an ISPF table is referenced, it must be in memory. Third, ISPF tables do not necessarily have to maintain the same number of variables (or columns) with each row. Fourth, ISPF tables may be keyed by one or more variables.

The requirement to maintain ISPF tables in memory does restrict the size of any particular table to that amount of memory which is available to an individual user. This requirement is, however, not too severe to the SAS application developer, because there are several SAS procedures which have a requirement to maintain an entire SAS data set in memory for processing.

The development for this solution hinges upon the altering of the data structure of the SAS Usage Notes. Figures SUGI-019, SUGI-020, and SUGI-021 describe the relationship between the SAS Usage Note data set and the ISPF table structures. This ISPF dialog approach would propose to break apart the Usage Notes into tables DMISUNT, DMISUNK and DMISUND. The CONTENTS of the SAS data set DMISUND, DMISUNK and DMISUND are in the figures before they were converted to ISPF tables.

The DMISUND table corresponds the Usage Note titles as seen in Figure SUGI-014. In this table there is a one to one correspondence to the observations in the Usage Note data set. The DMISUND table corresponds to the text of the Usage Notes. In effect, the TEXT1--TEXT20 variables in the Usage Note data set were transposed into observations to form the ISPF table. The DMISUNK table is a many to mapping from the Usage Note variables which relate to the know problems, releases, and the KEY variable from the Usage Note data set.

In Figure SUGI-014 there is a line in the table display labeled "Subset based upon:" Entries in these fields are matched against the DMISUNK table. If matches are found then the titles which correspond to the matches are displayed from the DMISUND table.

Using the standard keywords within the Usage Notes data set would could easily "Search" for notes based upon multiple searching criteria. Figure SUGI-019a demonstrates a sample of an inquiry to find known LIMITATIONS in the general SYS.SYS area of SAS.
If we selected the last title in the table display, we would see the text of the Usage Note which would come from the DMISUNT table as shown in Figure SUGI-020a. Using features unique to SAS/DMI would easily browse the ZAPS data set from this panel by entering the primary command of ZAPS.

Figure SUGI-021a demonstrates the coding involved with displaying the data and ZAP data set. It should be noted that there is SAS data step code controlling the interaction between the tables and managing the displays. There is no SAS coding necessary to browse the ZAP data set however. This is due to the use the &DMIEXEC variable directly within the PROC (processing section of the panel). &DMIEXEC is a VDEFINE user exit within SAS/DMI which permits the immediate execution of ISPF services.

Figures SUGI-024, SUGI-025, and SUGI-026 outline the pseudo code required to display the ISPF tables used within this dialog. Although the pseudo code masks some of the details, the fundamental concepts are to simply open the ISPF tables when required, display either the Usage Note Titles or Text, use the DMISUNK table to subset the Usage Note titles table.

Conclusion

The proposed application problem here demonstrated three solutions to a hypothetical problem. The acceptance by the application user would be affected by the solution which was chosen. Which solution that would be suggested is simply a question of the application user requirements. For many users the SAS/FSP solution is quite adequate. For others the SAS/AF solution, while others would require the SAS/DMI development.

Trade Mark references

SAS/FSP, SAS/AF and SAS/DMI are registered trademarks of SAS Institute Inc.
--- SAS Usage Note Titles ---
Command ==> Scroll ==> CSR

Subset based upon:
SCP Product Release Fixed Problem Module Usage Number

Usage Note Titles
- SAS Institute Usage Notes NOV86 disclaimer - Part I of II
- SAS Institute Usage Notes NOV86 disclaimer - Part II of II
- Recommended maintenance for SAS software products
- Maintenance for SAS/AF product
- Maintenance for DMS
- Maintenance for ETS Product
- Maintenance for SAS/FSP product
- Maintenance for SAS/GRAPH Product
- Maintenance for Base Product
- Maintenance for PROC PRINT
- Maintenance for BASE Product
- Maintenance for the SAS Macro Facility
- Maintenance for the SAS Macro Facility
- Maintenance for the SAS Macro Facility
- Maintenance for SAS/FSP

--- SAS Usage Note Titles ---
Command ==> Scroll ==> CSR

Subset based upon:
SCP Product Release Fixed Problem Module Usage Number

Usage Note Titles
- Explicit variable lengths taken modulo 256
- XEROX 8700 printers require manual intervention for error messages
- Maximum of 32,767 input lines

--- SAS Usage Note ---
Command ==> Scroll ==> CSR

MODULE=SYS.SYS-V52220 SYSTEM=OS PRODUCT=SAS RELEASE=5.08 FIXED=5.15
AVAILABLE ZAPS=Z5082220
SAS 5.08 still has the limitation of 32,767 input lines.
It will print the message 'ERROR: INPUT EXCEEDED MAXIMUM LINE COUNT.'
and abend with a U999 code. This limitation has been removed for SAS 5.15.

--- SAS Usage Note ---
Command ==> Scroll ==> CSR

MODULE=SYS.SYS-V52220 SYSTEM=OS PRODUCT=SAS RELEASE=5.08 FIXED=5.15
AVAILABLE ZAPS=Z5082220
SAS 5.08 still has the limitation of 32,767 input lines.
It will print the message 'ERROR: INPUT EXCEEDED MAXIMUM LINE COUNT.'
and abend with a U999 code. This limitation has been removed for SAS 5.15.
Program Logic for Usage Note Dialog

Note: Pseudo code used.

DATA _NULL_;  
initialize variables; 
TBOOPEN tables; 
DO WHILE (1) :  
TBDISPL DMISUND PANEL(OMISUND);  
IF ISP RC EQ 0 THEN GOTO DONE;  
IF TSEL EQ 'S' THEN LINK SHOWTEXT;  
IF WRKCHO EQ 'YES' THEN LINK SUBSET;  
END;  
DONE: ; TBEND tables; STOP; 
SHOWTEXT: ; scan for note and display 
RETURN; 
SUBSET: ; build TBSARG condition list for subset 
TBSARG DMISAST NAMECOND(USENOTE,EQ);  
TBSARG DMISAST ;  
DO WHILE (ISP RC EQ 0); 
TBDISPL DMISAST PANEL(DMISAST); 
END; 
RETURN;  
SUBSET: ; IF all WRKvars EQ ' ' THEN DO;  
... reset DMISAST titles;  
DMISAST = 'ALL';  
END; ELSE DO;  
... copy WRKvars to SAVvars;  
... build NAMECOND pair for Subset;  
TBSARG DMISASK NAMECOND(NAMECOND) ;  
TBSCAN DMISASK ;  
DO WHILE (ISP RC EQ 0); 
TBGET DMISASK ; 
DMISASK = 'Y';  
TBPUT DMISASK;  
TBSARG DMISASK ; 
END; 
DMISASK = 'SCAN';  
TBSARG DMISASK NAMECOND(DMISASK,EQ) ;  
END; 
RETURN; 

Reference Sources
- SAS/FSP User's Guide
- SAS/AF User's Guide
- SAS/DMI User's Guide
- ISPF Dialog Management Services
  Version 2.2 
  Publication Number SC34-4024
- ISPF/PDF Services
  Version 2.2 
  Publication Number SC34-4023
- ISPF Dialog Management Services SRA 
  Self Study Course 32302 
  IBM Education Course
- Full-Screen Testing of Interactive 
  Applications 
  M. E. Maurer 
  IBM Systems Journal 
  Vol 22, No 3, 1983