ABSTRACT

Many employees perceive and conduct the tasks related to data input, data management, analysis, and reporting as independent activities. This comes as no surprise when responsibility for these tasks is divided among several people and different products (such as SAS/FSP® and Base SAS® software) are required. SAS/EIS® software can be your solution to consolidate, organize, and simplify your business process making it easier to pass data between tasks, reducing system operation errors, and speeding up report generation. This paper demonstrates how SAS/EIS was used with existing hardware to build upon a SAS/FSP data entry system and to facilitate data file management, data validation, report generation, upload, download, and update operations. The key to successful implementation for this application was that with SAS/EIS we were able to mimic the business process by stepping through the tasks in the sequence required to meet data flow and reporting needs rather than simply presenting the SAS user with a set of individual SAS programs that required regular changes and execution at the proper time.

THE BUSINESS PROCESS

Unfortunately, some Commercial Credit customers do not repay their real estate loans. Surely all finance companies face similar problems. When this happens, the Commercial Credit branch manager fills out a foreclosure form with information about the initial loan, payment history, current expenses, present worth of the property, and current net balance. The branch manager sends the foreclosure form to the Home Office where the information is studied and a decision is made whether to initiate procedures for repossession, to consider the property a loss, or to reevaluate the situation in sixty days.

It is Helen-Jean's job to manage tracking and reporting activities related to these foreclosure forms. Several issues are important. Have branches prepared and submitted foreclosure forms for all real estate loans in default for sixty or more days? What is the reason that the customer has not made payments? Who approved the loan initially? Are the write-downs for the property appropriate considering present worth of the property, expenses, and net balance? What decision was made by Home Office? Has the branch followed through with the collection or repossession actions specified by Home Office? What is the current disposition of properties that were classified as a loss?

A few years ago Helen-Jean developed a SAS/FSP data entry system to collect pertinent information from the foreclosure forms and a set of SAS programs to address the issues mentioned above. This system will be called "our SAS/FSP system" for simplicity. Data flow and reporting requirements have not changed since then. Branch managers send foreclosure forms throughout the month with the greatest concentration at the end of the month. All forms for a particular month must be received by the 25th day of the month. Three people enter data on a networked personal computer using the SAS/FSP screens which validate values, automate calculations, and import data from a branch data set located on the network common drive. When all the data for a month have been entered (no later than the last day of the month), a copy of the data set is moved to another personal computer and the data entry computer is reset so that data from foreclosure forms can be entered for the next month.

Data validation and reporting begin once the data have moved to the second personal computer. Data validation programs detect observations with incomplete information and check for duplicate observations (since the same application could have been entered by different people). Errors are corrected using SAS/FSP. Preliminary reports then are prepared for Home Office review showing all the accounts represented by the foreclosure forms. Any corrections or additions are made using SAS/FSP. A second set of reports then is prepared to send to the branches for their review. Branch managers must return corrections no later than the fifth day of the month. Any further corrections are made using SAS/FSP. A final set of reports then is prepared to be sent to the branches and final data sets are created for uploading to the mainframe. Monthly production programs on the mainframe match foreclosure data from this and prior months with current information on more than two million accounts in the company to determine the disposition of accounts from the foreclosure forms and to determine whether all required foreclosure forms have been submitted.

Although our SAS/FSP system was very effective, it was dependent upon manual processes which had to be performed correctly in the right order. Operations varied considerably, ranging from uploading and downloading files, entering and correcting data, and backing up files, to running programs for reporting, data validation, and data management. Programs had to be performed in the correct sequence. Programs also had to be modified each month to read and to write the correct files because data set names changed each month.

There were several steps in our SAS/FSP system where operation errors occurred resulting in erroneous reports, program runs, lost time, and frustration. The errors were more frequent whenever there was a new employee operating the system. SAS/EIS software seemed to possess the tools needed to enhance our SAS/FSP system without much effort so that system operation errors could be avoided. Additional benefits would include faster report generation and reduced training time.

REENGINEERING THE APPLICATION

Because our SAS/EIS system was to be an enhancement to our SAS/FSP system, we had three main requirements: (1) to use existing hardware, (2) to produce an application that was easier to use than our SAS/FSP system, and (3) to expend minimal time and effort. The existing hardware consisted of mainframe computer (SAS version 6.08) and networked personal computers (SAS version 6.08 running Base SAS software and SAS/FSP). SAS/CONNECT software was used to transfer SAS data sets between the mainframe and personal computers. The limitation on time and effort meant rapid development. It was important to reuse as much code as possible for reports, validation routines, and data entry screens.

Because our SAS/FSP system had been operational for a few years, we knew where the difficulties lay. The primary objective for development of our SAS/EIS system was to reduce system operation errors, and secondary objectives were to speed up
report generation and to reduce training time. Most difficulties would be solved by developing a system that would make routine changes to programs and then run programs and other operations in the correct sequence. This meant that our SAS/EIS system needed to mimic the business process in terms of the sequence of operations in order to meet data flow and reporting needs.

Figure 1 shows a sample SAS/EIS system. It performs many useful tasks such as generating reports, producing graphs, and evaluating critical success factors. This approach is procedure-oriented and requires the user to understand what operations to execute and in what order. This approach did not offer much improvement over our SAS/FSP system.

Figure 1

Our SAS/EIS system needed to be process-oriented rather than procedure-oriented. We grouped operations according to data flow and reporting needs throughout the month: PRELIMINARY, INTERMEDIATE, and FINAL. The Preliminary group included activities on the data entry personal computer, starting with preparing the data set at the beginning of the month and ending with copying the data to the network at the end of the month. The Intermediate and Final groups included activities on the second personal computer. The Intermediate group started with data validation and ended with the first set of reports for the branches. The Final group started with final data corrections from the branches, and ended with final upload operations. Each of these groups contained different procedures. Figure 2 shows the desired design.

Figure 2

Our SAS/EIS system is the result of a joint venture between Helen-Jean and Alberto. The most important aspect of system design was for us to identify that the EIS structure needed to be process-oriented rather than procedure-oriented. Details related to using SAS/EIS objects in our SAS/EIS system are presented in the next section.

RAPID APPLICATION DEVELOPMENT

SAS/EIS software (version 6.08) contains 27 pre-written objects that build and execute SAS/EIS applications to perform particular functions. These objects are very easy to use for system development because you simply fill in the blanks and select desired items from lists to supply needed information to the object. Routines and data needed for the task are built into the object. This makes system development very fast because you can focus on system operation and function rather than on syntax.

We built our SAS/EIS system using several SAS/EIS objects such as DESKTOP, EXECUTE, LISTMENU, SCRIPT, EDIT, SIGNON, and LETTER. The main menu (Fig. 3) is a DESKTOP application. Preliminary, Intermediate, and Final activity groups are represented by graphics showing a leaf cutting, a small rooted plant, and a mature tree. Users may click on the graphic or the label below the graphic for each activity group. The graphic of
open hands represents help information. Ad-hoc reports and System exit also are available from the main menu.

A DESKTOP application enables the user to choose an application by selecting a region on a graphical menu. To build a DESKTOP application, you must complete the Graphics Menu Builder window to assign a name for the application. The Graphics Menu Builder window for the "MAINMENU" DESKTOP in our SAS/EIS system is shown in Figure 4. From the Graphics Menu Builder window select "Build" which opens the Build window. Here you use the SAS/EIS region manager to define objects on the DESKTOP surface and then fill the objects with graphic images or text. Our SAS/EIS system used stored graphic images for the plants and hands images, SAS software supplied icons for the Ad-hoc and Exit images, and text for the title and labels.

Each of the graphic images, icons, and associated labels has a target application identified which determines what happens when the user clicks on that area of the DESKTOP. When "Preliminary" is selected, the target application is a LISTMENU application called "PRMENU" and a selection list appears as shown in Figure 5.

A LISTMENU application enables the user to select a SAS/EIS application from a list that appears as either a scrollable list (of text or icons) or a popup menu. The LISTMENU can include a very large number of choices. To build a LISTMENU application, you must complete both the List Menu and the Edit Menu Text windows. In the List Menu window you assign a unique name for the application, provide a description and title, designate the window position, and specify whether the choices will be listed as scrollable text, scrollable icon, or popup menu. The List Menu window for the "PRMENU" LISTMENU in our SAS/EIS system is shown in Figure 6.

In the Edit Menu Text window of the LISTMENU object, you list each item for the selection list and specify the target application for each item. The first Edit Menu Text window for our "PRMENU" LISTMENU is shown in Figure 7. There is one of these windows for each selection in the list of choices for our "PRMENU" LISTMENU. The target application for this first choice is the "PREFRESH" EXECUTE application.
An EXECUTE application enables the user to execute SAS code that is stored in an external file or in a SAS catalog entry. To build an EXECUTE application, you must complete the Execute a SAS Program window to assign a name for the application. Then you select one of three secondary windows to indicate where the SAS program is stored. The Execute a SAS Program window for our "PREFRESH" EXECUTE is shown in Figure 8.

The secondary window chosen for our "PREFRESH" EXECUTE was the External File Name window (Fig. 9). We used this window because we wanted to specify the physical name for the SAS program. We would have selected a different secondary window if we had chosen to use a logical name for the SAS program file or if the SAS program had been stored in a SAS catalog entry. The file specified in this window, c:\cc698\programs\refresh.pgm, contains the SAS program to reset the data entry computer at the beginning of the month. This was one of many programs that we reused from our SAS/FSP system.

The target application for the second choice in our Preliminary activity group, "Data Entry" is the "PENTRY" EDIT application. An EDIT application enables the user to edit or to browse a SAS data set using SAS/FSP. To build an EDIT application, you must complete the SAS Data Set window to assign a unique name for the application and to specify the name of the data set that will be edited or browsed. The SAS Data Set window for our "PENTRY" EDIT (Fig. 10) specifies the name of the data set, IN1.FCREPO, for the first data entry screen in our SAS/FSP system. The type of access for the data set is Edit and one record is displayed at a time.

We also selected the Custom Catalog secondary window so that we could specify the location of the SAS/FSP screen catalog entry to start our data entry system. This is another example of our SAS/EIS system reusing code and screens from our SAS/FSP system.

Reports are produced in the Intermediate and Final activity groups of our SAS/EIS system. From the scrollable list shown in Figure 11, the user can choose to run the first set of intermediate report programs individually, in which case EXECUTE applications are invoked, or the user can choose to run all of the reports as a group, in which case a SCRIPT application is invoked.
A SCRIPT application enables the user to execute a group of SAS/EIS applications in sequential order. To build a SCRIPT application, you must complete the Script of SAS/EIS Applications window to specify a unique name for the application and to enter the names of the SAS/EIS applications to execute. The applications will execute in the order listed. The Script of SAS/EIS Applications window for our "ALL11" SCRIPT is shown in Figure 12.

**Figure 12**

**A SIMPLE CLIENT/SERVER SOLUTION**

SAS data sets are uploaded and downloaded to and from the mainframe computer in the Intermediate and Final activity groups of our SAS/EIS system with the help of a SIGNON application. A SIGNON application enables the user to either establish a connection or terminate an existing connection between SAS sessions on two computers. This application uses SAS/CONNECT software. To build a SIGNON application, you must complete the Remote Connect window to specify a unique name for the application, to identify the remote session which you want to establish or terminate, to specify the name of the file containing the SAS/CONNECT signon or signoff script on your system, to identify the communication access method, and to choose whether to establish or to terminate the communication link. The Remote Connect window for our "CONNECT" SIGNON application is shown in Figure 13.

**Figure 13**

**PROVIDING HELP FOR THE USER**

We used a LETTER application to provide the users help information which describes what work is done in each activity group along with some notes about special cases to be aware of and some general warnings. A LETTER application enables the user to access or to execute a FSLETTER application to browse, to edit, to send, or to print LETTER catalog entries. To build a LETTER application, you must complete the Letter Application window to specify the name of the application and to identify the type of access to the LETTER catalog entry (choices include Browse, Edit, Print, and Send). From this window you then select a secondary window, Letter Catalog, to specify the libref, catalog, and entry name for the LETTER entry to be accessed by the LETTER application. Figure 14 shows the Letter Application window for our "HELP" LETTER followed by a portion of the user help (Figure 15).

**Figure 14**

1010
CONCLUSIONS

SAS/EIS software provided an easy solution to consolidate, to organize, and to simplify the tracking of foreclosure applications in the Commercial Credit Home Office. We satisfied the requirements for application development of using existing hardware, producing an easy-to-operate system, and expanding little time and effort. We also meet the goals of reducing system operation errors, speeding up report generation, and reducing training time. The key to successful development of our SAS/EIS system was that the EIS was designed to mimic the business process by stepping through the tasks in sequence required to meet data flow and reporting needs rather than simply presenting the user with programs and procedures that require regular changes and execution at the proper time. The other important factor was that the SAS/EIS system built upon an existing SAS/FSP system and we were able to reuse program code and screen catalogs.

REFERENCES


ACKNOWLEDGMENTS

SAS, SAS/AF, SAS/CONNECT, SAS/EIS, and SAS/FSP 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 or trademarks of their respective companies.