INTERACTIVE SAS USE IN BATCH MODE: MAKING THE SAS SYSTEM USER FRIENDLY IN A ROSCOE BATCH ENVIRONMENT

S. David Riba
E-Systems, Inc.
Communications Manufacturing Division

ABSTRACT

While there has been much discussion about the use of SAS in an interactive environment such as TSO, users of SAS in batch mode have a variety of problems that need to be addressed. At E-Systems we have developed several methods that have proven effective in minimizing these limitations. This paper will review our experiences with modular program design, ROSCOE RPF, and in-house SAS support.

INTRODUCTION

E-Systems, Inc. is a major supplier of sophisticated defense electronics with headquarters in Dallas, Texas. Three divisions of E-Systems, Communications Manufacturing Division, ECI, and Memcor, are located in Florida and share the same computer mainframe. This mainframe is an IBM 3083 with MVS, TSO, ROSCOE and SAS software. SAS is accessible to end users only in batch using ROSCOE, a text/editor product of Applied Data Research, Inc.

Over the past 18 months, there has been a significant effort to distribute data access from the MIS Department to the end user. Through the training of over 400 employees in the use of ROSCOE and SAS to generate data analysis and management reports, end user capabilities have increased dramatically. Our SAS users at E-Systems range from the novice to the experienced programmer, with applications that range from the business environment of finance, purchasing, manufacturing and human resources, to scientific requirements in engineering and statistical analysis. We have developed a variety of techniques to minimize some of the inherent limitations imposed by this operating environment.

MODULAR SAS PROGRAM DESIGN

New users at E-Systems generally have no prior computer experience. Training begins with an overview of the computer, review of the computer terminal and keyboard, and an Introduction to ROSCOE class. The new user leaves this introductory session with the ability to access the computer, manage some simple files, and perform some basic text editing commands.

Once our users have become familiar with ROSCOE, they are then introduced to SAS. In our SAS training we emphasize a modular concept of SAS program design: that is, while all SAS programs consist of JCL, data definition, data, data analysis, and report writers, we stress that these elements be developed and maintained as separate modules to be chained together only when a particular job is ready for submission.

There are several advantages to this method of SAS program design.

First, it is easier for the novice to develop single purpose modules than to try to create the entire program at one time. This also helps with the problem of remembering to differentiate Data Step instructions from Proc Step instructions.

Second, it allows a significantly higher degree of flexibility in program design. Since the various elements can be easily interchanged to accommodate varying needs. Thus several different report writers can be created using the same data and analysis modules; or, by replacing the data analysis portion of the program, an existing modeling program can easily become a forecasting program using the same data and report...
format. Again, by replacing the data module, the same analysis can be useful in a sister division with only minor modifications, thus allowing the economies of pooled resources.

Third, creation of program modules allows a higher degree of interdependence among our analysts, allowing them to “mix or match” program elements to meet their requirement. This minimizes the need to “recreate the wheel” for each new application. This process creates a greater degree of confidence in the quality of the data. All analysts working on a common application and using the same data will reduce reporting errors caused by incomplete or obsolete data.

For example, our financial analysts might develop a SAS model consisting of:

- Data set of information extracted from division accounting data.
- Analysis program on historical information.
- Projection analysis module.
- Several different report writers.

These elements could be combined in various ways as needed. A clerk would be assigned to maintain the data and keep it current. Several analysts could then access this data to generate desired SAS programs without the time or cost of using an analyst to maintain data files. In addition, since our other divisions may have similar requirements, the modular design of this application would be readily useful merely by substituting the data used and modifying the report writers.

**ROSCOE RPF INTEGRATION**

As our users become more proficient, we then incorporate some of the more advanced features of ROSCOE to enhance the modular programming concept. ROSCOE has a feature called RPF (ROSCOE Programming Facility) which allows for the creation of screen displays and the generation of ROSCOE instructions. This can be helpful in specifying program options between various modules and in making modifications to the resulting SAS program.

In the case of SAS programs written for novice users or those not interested in the mechanics of SAS, RPF programs can assist in some of the more mundane tasks, such as modifying dates within the report writers and making selections between a variety of options. In practice, the financial modeling application mentioned above might be enhanced using RPF inputs to select:

1. Whether the report is to be historical or a projection.
2. Dates of data to be included or excluded.
3. Other key elements to be included or excluded.
4. Which reports are to be generated.
5. Distribution of the reports.
6. JCL under which the report is to be submitted.

The RPF program would then selectively comment or uncomment key portions of the SAS program modules, chain the selected program modules together, add the appropriate JCL, and then submit the program for execution. A simple RPF program might look like this:

```
REPORT
FETCH EX.CMD0000
DEL 2 3
COPY JCL 1. 1
COPY CX.CMDABC1 1.. B
COPY CX.CMDXYZ 1.. B
EDIT \WKEND=MDY(00.00.81);/WKEND=MDY(03.16.81);
SUBMIT THIS JOB
```

RPF panels can be used to build SAS data files since they can structure the data input and test each element for validity before accepting the element as data. Every field can be tested for length, accuracy, and conformity to preselected criteria. Thus, a date field can be verified for proper month, day, and year. The date can also be tested for a preselected range of valid dates. In the case of prices, the field may contain a dollar sign and period in the proper places, but be verified entirely numeric otherwise.

An RPF-driven program can be made to appear interactive to the user through the use of a timed loop to retrieve job output information and redisplay the data on the screen for viewing or further analysis. A SAS job can direct the output to a disk or tape file which can then be retrieved and the data redisplayed to the user for review. If desired, the user can then resubmit the SAS job with revised data. This approach must be used with caution since it is important to include a time limit on the RPF program. Waiting for a batch
SAS job to complete execution may cause a potentially endless loop.

```plaintext
<<CHEK>>
LET PI = 0
<<LOOP>>
TRAP ON
LET PI = PI + 1
WAIT 10
IF PI EQ 10
      RETURN
ENDIR
<<BYTE>>
ENDOF
LET Pl .. Pl ... 1
WAIT 30
IMP DSN.TB.ABC.CMD1234
IF Pl EQ 10
      GOTO <<BYE>>
ENDIr
11" RC NE 0
GQTO «LOOP»
ENDOF
RETlJRN
```

In practice, an RPF driven interactive SAS session might perform the following functions:

1. By accessing the RPF program, the user would see on the screen a list of available options for this program. These options could range from creating a SAS data set to submitting a SAS job.

2. Based on the options selected, the user would then be guided through another preselected range of options. If the option selected were to build a SAS data set, the panel would accept the inputs for the data set, verify them against established criteria, and write the results into a library number for use. If the option selected were to issue a report, the panel would prompt for options to be included or excluded from the report and edit the SAS program accordingly. Thus, a part number may be selected for inclusion or exclusion, a range of dates may be specified, a fiscal period for the report may be selected.

3. A temporary SAS program would be created by chaining the required program modules together as specified by the RPF program (e.g., Report Writer B instead of Report Writer A, Analysis C instead of Analysis A, etc.). This temporary SAS program would then be modified based upon the panel prompts from point 2.

4. The user's Job Control Language would be added and the job would be submitted to the computer for execution.

5. A timed loop would idle the RPF program until the SAS program had completed running.

6. Once the SAS program was finished, the results could be imported into the user's library and the panel could be reloaded and redisplayed.

The user could then be given the option of reviewing the information from the completed job, modifying the data and resubmitting a new job, or printing the reports.

7. These steps could be repeated as often as necessary.

This approach to developing SAS programs creates a fairly high level of end user interdependence. It also significantly increases the speed with which novices start writing more sophisticated programs. Because the problem is now reduced to simpler elements, writing SAS programs becomes a much less ponderous task for the novice. In our applications, we have found few unique SAS problems that were too specialized to have relevance to our other SAS users. Generally, we found that many problems could be resolved by modifying an existing SAS program module created by another user.

IN-HOUSE SAS SUPPORT

Key SAS users are identified at the local level in each division. They become the resources for users with problems or questions. If these key users are unable to help, each division is assigned an individual SAS coordinator to assist with system design, user programming support, and training. This network of support personnel fills the void in user support that exists under more traditional data processing organizations. Within the user's own area, responses become more timely and supportive. In addition, new techniques and other information are easier to disseminate through the key users, since they are aware of those who can best use the information.

For additional information on using SAS in a RASCON batch environment, contact the author at:

E-SYSTEMS, INC.
COMMUNICATIONS MANUFACTURING DIVISION
P. O. BOX 33010
ST. PETERSBURG, FLORIDA 33733
(813) 381-1155