AN INTERACTIVE METHOD TO INVOKE SAS REPORT PROGRAMS

Jon L. Bond
IBM Corporation
Boulder, CO 80301-9191

Abstract: This paper describes how a customer can enter parameters via the Interactive System Productivity Facility (ISPF) under the Multiple Virtual Systems / Extended Architecture (MVS/ESA) operating system, causing a Statistical Analysis System (SAS) job to be submitted to create SAS reports.

ISPF routines allow a menu-driven interface to receive parameters, which are eventually passed on to the modified SAS programs. In this application, previously written SAS programs are modified to include SAS macro-language statements.

This paper focuses on the general methodology needed to use this technique.

Implementation of an Interactive Method

MVS/ESA contains the basic elements of information that describe the field performance of a machine. A method was needed, by which a customer, usually a nonprogrammer, could pass variables to SAS programs. Using an IBM licensed program called Interactive System Productivity Facility (ISPF), the customer can enter into a dialog with the system and construct the JCL required to execute an existing SAS program. A dialog is an interactive application designed to run under the control of ISPF, which manages interactive applications. ISPF provides control and services to permit execution of the dialogs in the MVS/ESA environments.

Two actions necessary to implement this new approach are as follows:

• Modify SAS code to include SAS macro variables.
• Code certain ISPF elements to provide an interactive interface.

Through careful design it is possible that multiple ISPF elements (such as panels, messages, and skeleton data sets) can be used for a number of the reports selected.

Modification of SAS code

Because many of the report requests involve making minor changes to programs, the main objective is to utilize existing SAS programs. Use of the SAS macro language allows the ISPF parameters to be passed to these slightly modified SAS programs, enabling reports to be generated.

Basically, an existing SAS program is rewritten to accept SAS macro variables that are set according to customer prompts, using ISPF. Each existing program is converted by removing SAS variable names and replacing them with macro variables (sometimes called symbolic variables).

```
SET Y86DD.SMAIL Y85DD.SMAIL;
IF DATE = '860303' THEN
  IF PART = '45362718' THEN

Original SAS Code
```

Introduction

Monthly reports distributed by the Field Data department contain considerable information at a generalized level. However, many customers (individual product areas) often need specific, detailed information. A new technique, called the interactive method, allows a customer to generate that information.

Prior to this interactive method, the Field Data department received many requests from customers. These requests were categorized as follows:

• Parts analysis
• Individual machine summary
• Parts usage summary
• Early-life information.

Department programmers would then make job control language (JCL) changes and SAS program modifications to retrieve the requested data in a report format. Although there were basically only four types of requests, processing individual requests for a report was time consuming because the requests would differ, for example, on product type and inclusive dates.

A typical scenario occurred when a customer called someone in the department and asked for a report. The person that received the call made coding changes and JCL modifications to a SAS program. Then a job was submitted to MVS/ESA, which resulted in a detailed customer report.

Previous Method. Department programmers "caught in the middle."
A Report Generation Dialog under ISPF

SAS Macro Processor

After the JCL is submitted as a batch job, the SAS program is executed under the MVS/XA system. The SAS macro processor then executes the macro operations. In this application, a SAS symbolic variable is defined, using the form &ALPHA, where ALPHA is a character string built by ISPF. When the SAS code recognizes this symbolic variable preceded by an ampersand (&), the macro processor replaces this symbolic-variable reference with the character string built by ISPF. The code is then passed directly to the SAS DATA and the PROC step processors.

Existing SAS programs may now be submitted via ISPF to an MVS/XA system. By using ISPF and the SAS macro processor in this interactive technique, the customer can be prompted for information and job parameters. The JCL to access the correct data sets is generated, and a background job is submitted to invoke the previously written SAS job.

The customer needing more-detailed information can obtain existing SAS reports quickly. Using this technique minimizes efforts to obtain detailed reports and removes department programmers as intermediaries in the report process.