Paper 95-25

Job Control Language and the SAS® System for Beginners

Robert J. Nowosielski, PECO Energy, Philadelphia, PA

ABSTRACT

SAS under IBM's MVS environment is a powerful data manipulator. But for the novice user, learning and understanding the methods for getting data into and out of the SAS environment can be frustrating. This is especially true in batch mode even though batch mode is often the best way to manipulate external data.

This paper explains the relationship between the SAS environment and the MVS environment in terms of files, DD names and Job Control Language (JCL). The process of getting data into and out of the SAS environment in JCL is explained in simple but descriptive terms, using both real world examples and textbook definitions. The novice will gain a basic understanding of accessing and manipulating both SAS and external files using JCL in conjunctions with the SAS Cataloged Procedure.

Keywords: JCL, DD NAME, JOB CARD, SAS Cataloged Procedure, FILENAME

INTRODUCTION

Many users of the SAS system under MVS never have a need to leave the SAS environment to develop, save, and execute their SAS programs. The SAS interactive environment under TSO allows them to interact directly with the operating system and be successful in most of their programming tasks. SAS provides all the tools necessary to read, write, and create system files, SAS Databases, and reports. There are many times however, when the ability to run SAS in a batch or background mode is better suited to the immediate task. When reading large databases or files, or when a production program will take a long time to execute, a background job can free the SAS programmer to do other things while programs are executing. SAS programs that are executed in the background can be scheduled to run automatically during off peak usage times. Background jobs are a convenient way to prepare large files and databases for further analysis. Background jobs can be part of a comprehensive data capture application where transactions are created and stored on-line and queued for update at a later time.

The command language used under MVS for executing background jobs is called Job Control Language or JCL.

JOB CONTROL LANGUAGE (JCL)

JCL is the language that allows the programmer to interact with the MVS operating system in batch or background mode. It performs the tasks of:

- Identifying the job to the operating system
- Identifying all the input/output requirements for the job
- Identifying where to find the programs that are to be executed
- Identifying when and in what sequence to perform the tasks requested.

Additionally, JCL allows the programmer to instruct the operating system to create, delete, or catalog files. It provides mechanisms to define file attributes and dispositions, and direct printed output.

JCL is submitted to the JCL processor on 80-byte records. Each JCL statement can consist of one or more records. Columns 1 and 2 of each JCL statement must contain the characters //. Remarks contain //* in columns 1, 2, and 3. A /* is a delimiter statement. Statements may be continued on one or more cards by placing a comma at the end of the card and then continuing the statement on the next card. The statements are in columns 3 through 71. Columns 71 through 80 may be comments.

The collection of all the statements is called a Job.

JOB STATEMENT

The Job statement is always the first statement in the job stream. It marks the beginning of the job and assigns the job a name. It provides a means for assigning accounting, programmer's name, the expected run time, and other general information about the job.

The format of the job card is

```
//jobname JOB accounting parameters comments
```

The jobname field can be any 8-character string beginning with a character. It identifies the statement so that other statements can refer to it.

The JOB field tells the system that this is a JOB statement.

The accounting field assigns accounting to the job.

The parameter control the way the job is scheduled, the resources assigned to it, and the manner in which output should be handled.

DD STATEMENT

The DD (data definition) statement describes a data set and specifies the input and output resources needed for the data set. The SAS system performs the similar tasks using the FILENAME and LIBNAME statements. The DD statements consists of the characters // in columns 1 and 2,a name field with a maximum of 8 alphanumeric starting in column 3, and the characters DD. A typical format is

```
//MYSAS    DD   DSN=MY.SAS.DBASE, DISP=SHR
//MYIN     DD   DSN=MY.INPUT.DATA, DISP= OLD
```

where DSN stands for Data Set Name and DISP stands for disposition. A file must be allocated before the SAS System can use it. After these statements are processed by the Operating System the data sets are allocated and are available to the SAS system for processing. It means that the Operating System has established a link between the physical data sets - HIGHLEV.MIDLEV. LOWLEV – and the DD Name – MYNAME. When a SAS program needs access to the data set, via an INFILE statement or, a SET Statement in a DATA Step, the data set can be referred to by referring to the DD Name. The following DATA Step, executed in a batch job will read the data stored in the external file called MY.INPUT.DATA and store it in the SAS data set called MY.SAS.DBASE.

```
Data MYSAS. MYCOPY;
File MYIN;
        Input Field1 Field2 Field3;
```

EXEC STATEMENT

The EXEC statement identifies the program the system is to execute for the current job. It marks the beginning of each step in a job (called a job step) and tells the system how to process the job step. The job step can include a STEP Name that allows the step to be referred to by other job steps or other statements.

```
//SAS  EXEC SAS
```

The programs that are most often used under MVS to execute SAS programs is the SAS CLIST for TSO Execution or the SAS Cataloged Procedure for batch processing.

SAS CLIST

The SAS CLIST is an external file containing TSO commands and control instructions. These commands allocate required files, select SAS data libraries, and determine many default parameters that are used during the SAS session. It enables a SAS foreground session under TSO to be executed with a single word.

SAS CATALOGED PROCEDURE

The SAS cataloged procedure is a collection of JCL statements needed to execute a program. It is packaged so all the JCL statements can be invoked in a single EXEC statement. The SAS cataloged procedure can be modified at execution startup by specifying parameters on the EXEC statement. The following parameters can be specified:

CONFIG= external file
Specifies the user configuration file that contains SAS system options that can override default values in the SAS standard configuration file.
ENTRY= entry point
The installation sets a default value for this that specifies the program module that is to be invoked for starting a SAS session.
OPTIONS= options
Specifies a list of SAS system options. These options are in effect unless reset in the SAS program.
WORK=n,n
Specifies the number of blocks of spaces to allocate for your WORK data library.

The following EXEC statement will replace the default value specified for the CONFIG= parameter with the system data set called MY.CONFIG.SAS.

```
//SAS  EXEC SAS,CONFIG='MY.CONFIG.SAS'
```

EXAMPLE

```
1//MYJOBA JOB 'MyAccounting','PGMRName',
 //   CLASS=B,NOTIFY=MYUSER,MSGCLASS=P
2//SAS  EXEC SAS,
  //   REG=8M,OPTIONS='LINESIZE=80'
3//SASLIST     DD SYSOUT=H,DEST=A0ZD
4//MASTAMT DD  DSN=MYDATA.DATASET,
 //      DISP=SHR
5//SASLIB  DD DSN=TD.SASOPS.LIBRARY,
 //        DISP=OLD
6//WORK DD UNIT=SYSDA,
 //        SPACE=(CYL,(200,50),,,ROUND)
7//SYSIN  DD *

        8DATA WORK.M;
        9           INFILE MASTAMT;
        10            INPUT FIELD1 $1-10
        11                   FIELD2  $11-20
                          ;
        12            IF FIELD = '33GCPM00CAP';
                    RUN;
         Proc Print;
         RUN
```

Line 1 makes up the JOB statement. It is continued on the next line by ending the line with a comma and beginning the next card with //. This JOB statement includes several parameters: Accounting is established by the installation Programmer Name is in quotes if it contains blanks Job Class identifies to the system the time and priority of the JOB and in many installations, it sets the maximum amount of run time allocated to the JOB Notification that allows the system to send a message to the user of the JOB if that user is logged onto TSO The MSGCLASS specifies how the system should route the output associated with this job. Each installation establishes the MSGCLASS during installation of the operating system.

Line 2 is the EXEC statement. It is continued on the next line. In this case, the SAS cataloged procedure is being executed. The SAS catalog procedure invokes the SAS system. This statement includes several parameters: REG=8M tells the system to allocate 8000 1024-byte (K) areas of storage to the SAS system while it is executing. The REGion parameter can be coded on either the JOB statement or the EXEC statement. OPTIONS='LINESIZE=80' is a SAS system option and is passed directly to the SAS system. This parameter sets the output line size to 80 characters.

Line 3 is a DD statement that establishes the SASLIST output file. It tells SAS where to send output generated by the SAS program. In this case, any output from the program will be sent to the system-established output queue called SYSOUT. The DEST parameter tells the operating system where to send any printed output.

Line 4 is a DD statement that allocates a file. This statement is continued on the next line. Using a FILENAME statement within the SAS program accomplishes the same thing. The file is allocated to the DD Name MASTAMT and can be referred to by a SAS program. The DISP=SHR parameter establishes that the file can be shared with other programs while this JOB is running.

Line 5 is a DD statement that establishes the SASLIB library for this JOB. The SASLIB file is used to store permanent SAS formats, informats and functions. The system looks here for before it looks in the system-established library. In this case, the library is allocated as DISP=OLD in order to allow the SAS system to write to the library if required by the SAS program.

Line 6 is a DD statement that establishes parameters for the WORK file that SAS will use. In this case, UNIT=SYSDA tells the system to use any disk storage device available. It also tells the system to allocate in 200 cylinders of space.

Line 7 is a DD statement that tells SAS where the program statements for the job are located. The SYSIN DD statement followed by the * means that the program statements will follow this statement.

Line 8 begins the SAS program that reads in 2 fields from an external file and writes them to a SAS work file and will print the data in the work file to the places established by the SASLIST DD statement on Line 3.

CONCLUSIONS

The SAS system under MVS can be executed in both foreground mode or batch mode. JCL is used to submit SAS programs to the batch processor. The JOB, DD, and EXEC statements are straightforward and easily learned. If you are running programs against large datasets or have a need to schedule your work to run automatically then learning JCL will be time well spent.

REFERENCES

SAS Institute Inc.,
SAS© Companion for the MVS Environment, Version 6, First Edition,
Cary, NC: SAS Institute Inc., 1990. 599 pp.

SAS Institute Inc
SAS© Language: Reference, Version 6, First Edition,
Cary, NC: SAS Institute Inc., 1990. 1042 pp.

Brown, Gary DeWard
System /370 Job Control Language,
John Wiley & Sons, Inc.,
New York, NY, 1970. 297 pp.

IBM Corporation
Program Product, MVS/Extended Architecture, JCL User’s Guide
GC28-1351-4
Fifth Edition, 1989

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Robert J. Nowosielski
PECO Energy
860 Ridge Pike
Plymouth Meeting, PA 19462-1945
Work Phone: 610-941-1551
Fax: 610-941-1673
Email: rnowosielski@peco-energy.com
Web: www.peco.com