

A SAS GENERATED BACKUP FACILITY

Barbara Belasco and Suzanne Ives, Sandoz, Inc.

1. ABSTRACT

User error and system crashes necessitate periodic backups of SAS data sets. Although the corporate computer operations at Sandoz backs up clinical data sets weekly, we, as system users, realized that to provide for data integrity we needed to backup data sets daily. Commercial backup software is costly and so we decided to use SAS to create the backups.

The automated backup facility implemented at Sandoz uses SAS to read the system catalog and generate the backup job stream. The UNIX operating system is used to automatically send the job to the IBM mainframe at a predefined time and to store its output in a prenamed file.

2. ENVIRONMENT

Clinical Research systems at Sandoz, Inc. operate under the constraints of FDA regulations which require data security and integrity. The clinical data management system which is used at Sandoz requires the daily changing of large numbers of data sets and the frequent addition of new data sets.

At Sandoz, all clinical files are stored in SAS data sets. They are stored on disk as SAS data libraries organized by compound identifier and study number. Each data library is named by a structured name in the following format:

```
RAD.CLINICAL.SAS.CCCSSSS
```

where CCCC is the compound identifier and SSSS is the clinical study number. The number of data sets which are active at a time can exceed 800.

The SAS data library backup procedure which is described below was developed because of the desire to automate the daily backup of data libraries so that the procedure does not require constant attention. In addition, it was desirable to execute the procedure automatically so that no one needed to remember to update the list of data libraries to be backed up nor to send the backup job.

The Sandoz clinical data management system runs on a DEC 11/70 computer which operates under UNIX. The DEC 11/70 is a front and back end processor to an IBM 4341. The SAS clinical data libraries are cataloged OS files on the IBM computer.

3. PROCEDURE

Figure 1 at the end of the paper illustrates the Sandoz backup generation system.

The UNIX operating system includes a program called "cron" which periodically checks the login directories of all system users for a file called .cron. The "cron" program executes the commands stored in the .cron file at the time and day specified. The backup generation job, which is stored in a file on the DEC 11/70, is scheduled to be sent to the IBM at 7:00 AM every weekday morning. A listing of the job is at the end of the paper in Figure 2.

The job first reads the system catalog. It selects entries for disk files whose names begin with RAD.CLINICAL.SAS. The program then uses PUT statements to create JCL and SAS code to create a new job to send to the OS internal reader to be executed. See Figure 3 for a sample of the job which is generated and sent to the internal reader.

The generated job is automatically executed from the internal reader and copies all of the cataloged disk files onto a tape. The printed output of the job is sent back to the DEC 11/70 where it is automatically stored in a disk in a particular directory. The file containing the output is named by the julian date of its execution. The Data Administrator can check the output for problems or can locate the number of the backup tape if it is needed.

4. PROBLEM

If a SAS data library has been allocated and cataloged but no data sets have been added to it, the library cannot be copied using PROC COPY. You can avoid this problem if you create a dummy SAS data set whenever you catalog a new library.

5. OTHER APPLICATIONS

The technique of generating OS jobs from SAS programs can be used for other applications where the JCL or SAS code is not known until run time or where human intervention to set up such jobs is neither desirable nor available.

6. ADDITIONAL INFORMATION

The authors may be contacted at Sandoz, Inc., Route 10, East Hanover, New Jersey 07936.

FIGURE 1: Backup Generation System

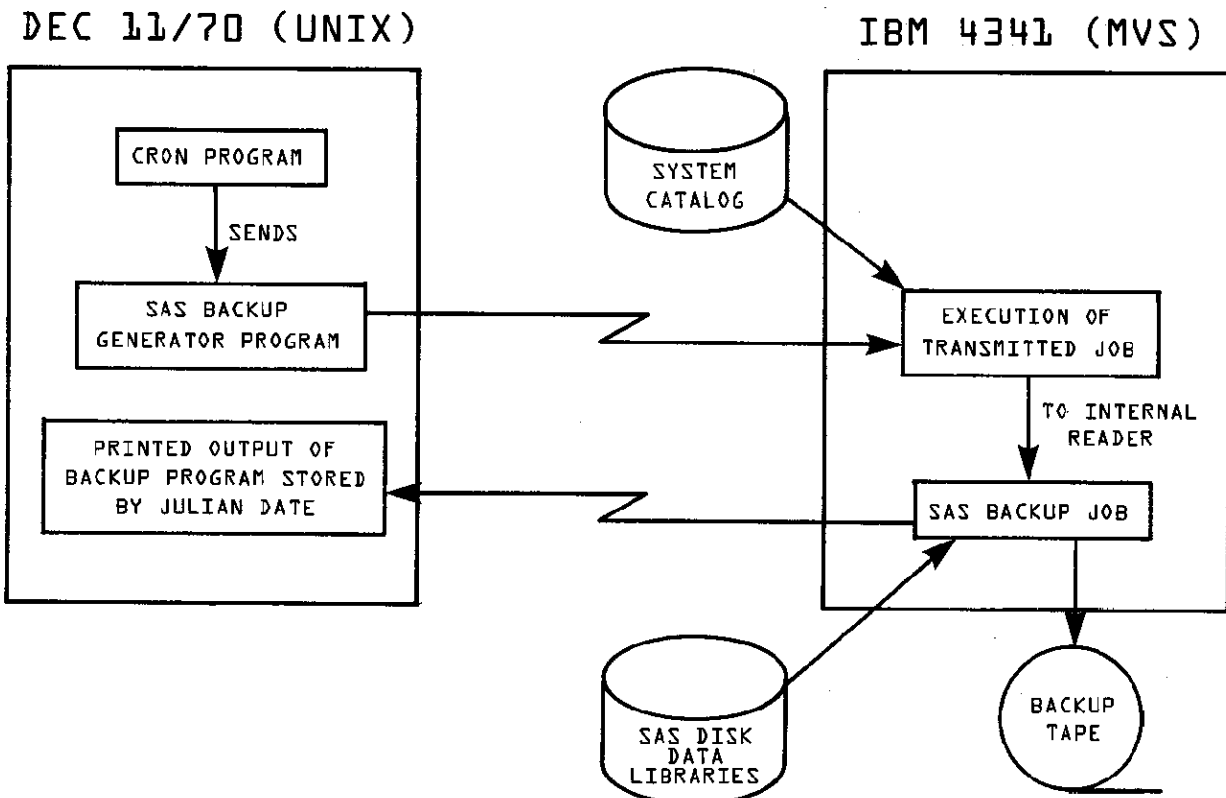


FIGURE 2: Backup Generator Job

```

//          JOB
//SIOF1     EXEC PGM=IDCAMS
//SYSPRINT DD DSN=&&IDCAMS,DISP=(,PASS),UNIT=DISK,
//          SPACE=(TRK,(5,5),RLSE)
//SYSIN    DD *
LISTC
          CAT(CATALOG.VSAND01) ALL
/*
//SAS      EXEC SAS,SYSOUT='*',COND=(0,NE)
//IDCAMS   DD DSN=&&IDCAMS,DISP=(OLD,DELETE)
//DISK     DD DSN=&&CATLG,DISP=(,PASS),UNIT=DISK,SPACE=(TRK,(5,5))
//SYSIN    DD *
DATA WORK1;
RETAIN DSN VOLSER UNIT;
KEEP DSN VOLSER UNIT;
INFILE IDCAMS LENGTH=LRECL;
  
```

```

INPUT @;
IF LRECL GT 8;          *TO FIND NONVSAM PRINT LINE;
INPUT @2 HEADER1 $CHAR7. @;
  IF HEADER1 EQ 'NONVSAM' THEN DO;
    HOWLONG = LRECL - 17;
    INPUT @18 DSN $VARYING44. HOWLONG;
    RETURN;
  END;
IF LRECL GT 60;
INPUT @9 HEADER2 $CHAR6. @;
  IF HEADER2 EQ 'VOLSER' THEN DO;
    INPUT @27 VOLSER $CHAR6.
      @53 DEVCODE $CHAR8.;
    IF DEVCODE EQ '3050200B' THEN UNIT = '3350';
    IF DEVCODE EQ '30008001' THEN UNIT = 'TAPE';
    IF DEVCODE EQ '32008003' THEN UNIT = 'TAPE';
    IF UNIT EQ '3350' OR UNIT EQ 'TAPE' THEN OUTPUT;
    DELETE;
  END;
DELETE;
DATA _NULL_;
SET WORK1;
FILE DISK;
PUT
  @1 DSN $44.
  @46 'UNIT = '
  @53 UNIT $4.
  @58 'VOLSER = '
  @67 VOLSER $6.;
//STEP03 EXEC SAS,SYSOUT='*'
//FT13F001 DD SYSOUT=(A,INTRDR)
//CATLG DD DSN=&&CATLG,DISP=(OLD,DELETE)
/* THIS PROGRAM GENERATED DAILY BACKUPS OF SAS DISK LIBRARIES */
DATA CATLG (DROP=COL1 COL7);
  INFILE CATLG MISSEVER;
  INPUT @53 U3350 $4.@;
  IF U3350 = '3350' ;
  INPUT @1 FILENAME $40.;
  IF FILENAME =: 'RAD.CLINICAL.SAS.';
DATA _NULL_ ;
  RETAIN SWITCH 1;
  FILE PUNCH;
  SET CATLG (IN=IN1) CATLG (IN=IN2) END=EOF;
  COL= -1;
  IF N=1
  THEN DO;
    PUT '//RADBACKUP JOB ' ; /*FILL IN JOB CARD INFORMATION*/
    PUT '//ROUTE PRINT RMT12';
    PUT '//STEPUSR EXEC PGM=IEBGNER';
    ~!date "+ PUT '//*** USR=(IVES,BACKUP/%j,54)';"
    /* "date" is a UNIX command which is executed when the job is
    sent to the IBM computer and which fills "%j" with the current
    date in julian format. "/ives/backup/julian date" is the
    name of UNIX file which receives the output of the generated
    job */
    PUT '//SYSPRINT DD DUMMY';
    PUT '//SYSIN DD DUMMY';
    PUT '//SYSUT2 DD SYSOUT=A,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)';
    PUT '//SYSUT1 DD *';
    ~!date "+ PUT '*** USR=(IVES,BACKUP/%j,54)';"
    PUT '//';
    PUT '//STEP01 EXEC SAS,OPTIONS='NOCENTER'';
  END;
IF IN1
  THEN DO;
    PUT '//FROM' _N_ 'DD DSN=' FILENAME +COL ',DISP=OLD';
    PUT '//TO' _N_ 'DD DSN=' FILENAME +COL ',DISP=(NEW,KEEP),';
    PUT '// LABEL=( '_N_ +COL ',SL,,EXPDT=99005),';

```

```

        IF N = 1
          THEN PUT ^// UNIT=(TAPE,,DEFER)^;
          ELSE PUT ^// VOL=REF=*.T01,UNIT=APP=T01^;
        END;
IF IN2 AND SWITCH = 1
  THEN PUT ^//SYSIN DD *^;
IF IN2
  THEN DO;
    PUT ^PROC COPY IN=FROM^ SWITCH +COL ^ OUT=TO^ SWITCH +COL ^;^;
    SWITCH=SWITCH+1;
  END;
IF EOF
  THEN PUT ^/*^ / ^//^;
/*
//

```

FIGURE 3: Sample Job Sent to the Internal Reader

```

//RADBACKUP JOB
/*ROUTE PRINT RMT12
//STEPUSR EXEC PGM=IEBGENER
/** * * * USR=(IVES,BACKUP/333,54)
//SYSPRINT DD DUMMY
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=A,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSUT1 DD *
** * * * USR=(IVES,BACKUP/333,54)
/*
//STEP01 EXEC SAS,OPTIONS='NOCENTER'
//FROM1 DD DSN=RAD.CLINICAL.SAS.COMP1,DISP=OLD
//T01 DD DSN=RAD.CLINICAL.SAS.COMP1,DISP=(NEW,KEEP),
// LABEL=(1,SL,,EXPDT=99005),
// UNIT=(TAPE,,DEFER)
//FROM2 DD DSN=RAD.CLINICAL.SAS.COMP2,DISP=OLD
//T02 DD DSN=RAD.CLINICAL.SAS.COMP2,DISP=(NEW,KEEP),
// LABEL=(2,SL,,EXPDT=99005),
// VOL=REF=*.T01,UNIT=APP=T01
//FROM3 DD DSN=RAD.CLINICAL.SAS.COMP3,DISP=OLD
//T03 DD DSN=RAD.CLINICAL.SAS.COMP3,DISP=(NEW,KEEP),
// LABEL=(3,SL,,EXPDT=99005),
// VOL=REF=*.T01,UNIT=APP=T01
//FROM4 DD DSN=RAD.CLINICAL.SAS.COMP4,DISP=OLD
//T04 DD DSN=RAD.CLINICAL.SAS.COMP4,DISP=(NEW,KEEP),
// LABEL=(4,SL,,EXPDT=99005),
// VOL=REF=*.T01,UNIT=APP=T01
//FROM5 DD DSN=RAD.CLINICAL.SAS.COMP5,DISP=OLD
//T05 DD DSN=RAD.CLINICAL.SAS.COMP5,DISP=(NEW,KEEP),
// LABEL=(5,SL,,EXPDT=99005),
// VOL=REF=*.T01,UNIT=APP=T01
//FROM6 DD DSN=RAD.CLINICAL.SAS.COMP6,DISP=OLD
//T06 DD DSN=RAD.CLINICAL.SAS.COMP6,DISP=(NEW,KEEP),
// LABEL=(6,SL,,EXPDT=99005),
// VOL=REF=*.T01,UNIT=APP=T01
//FROM7 DD DSN=RAD.CLINICAL.SAS.COMP7,DISP=OLD
//T07 DD DSN=RAD.CLINICAL.SAS.COMP7,DISP=(NEW,KEEP),
// LABEL=(7,SL,,EXPDT=99005),
// VOL=REF=*.T01,UNIT=APP=T01
//SYSIN DD *
PROC COPY IN=FROM1 OUT=T01;
PROC COPY IN=FROM2 OUT=T02;
PROC COPY IN=FROM3 OUT=T03;
PROC COPY IN=FROM4 OUT=T04;
PROC COPY IN=FROM5 OUT=T05;
PROC COPY IN=FROM6 OUT=T06;
PROC COPY IN=FROM7 OUT=T07;
/*
//

```