SCHEDULING AND MANAGEMENT OF PERIODIC REPORTS
Ross Z. Merlin, Pinkerton Computer Consultants, Inc.

ABSTRACT

Many data processing applications require that reports be produced at regular intervals -- daily, weekly, monthly, etc. To ensure that personnel reports are produced on schedule, a SAS based monitor system has been implemented at the Agency for International Development. This system generates the JCL for all reports that are due each day, submits the jobs, and advances the due-date as each report is completed.

The user can access the database to modify the due-date, destination, or number of copies for a particular report, as well as indicate which non-periodic reports should be submitted. SAS features illustrated by the programs include date-value manipulation, interactive transaction editing through terminal prompts, SAS/FSP, SAS programs generated by other SAS programs, and character manipulation (to build the JCL and SAS statements). Simplicity and flexibility are enhanced by utilizing JCL procedure parameters.

THE PROBLEM

With over 150 reports being produced at various intervals, there are too many opportunities for a job to be submitted on the wrong date, or to be skipped entirely. There is also the necessity to verify that each job has completed successfully, and there is the schedule to maintain so that the job runs again the next time it is due.

To make the problem a little more interesting, there are a few more requirements. Each job has the capability to produce a variable number of copies of up to four reports, and each job can call for up to three tape volumes. The user must be able to change the due-date, frequency, number of copies, and destination. JCL for the jobs must be kept in restricted production libraries, so that a change in the number of copies (for example) can not be made simply by editing the JCL.

THE SOLUTION

There are five components to the solution: a SAS database (the Monitor), JCL PROCs, a SAS job to submit all the jobs that are due, a SAS program to update the Monitor as each job runs, and a maintenance facility for the Monitor.

In the accompanying programs CPY4 and CLISTS, DDNAME "DB" refers to the Monitor database, which consists of five SAS datasets: MONITOR (the current version) and MONITOR1 through MONITOR4 (backup generations created by PROC DATASETS / AGE). RUNDATE refers to the processing date, which may lag behind the actual date. The Monitor dataset consists of the following fields: JOB, NAME, DUE, RPT1--RPT4, CPY1--CPY4, VOL1--VOL3, and KMT (see Figure 1). The three character JOB code links the record on the Monitor with the jobname and the JCL procedure (due to local requirements, all jobs and PROCs for this application must be prefixed with "0632"). FREQ is a one character code which specifies the frequency for the report, i.e. how often it is to be run: D=daily, W=weekly, B=bi-weekly, ... A=annually, X=as requested*. NAME is a brief description of the job. RPT1 through RPT4 are each three character codes which identify a particular report. CPY1 through CPY4 indicate the number of copies for the corresponding report in RPT1 through RPT4. The CPY and RPT fields become PARMs (parameters) for JCL PROCs. VOL1 through VOL3 are the VOLS (tape volume serial numbers). KMT is the remote printer identification.

The JCL procedure for job ABI is shown in Figure 2. In this example, there are three PARMs on the PROC statement: D03, VOL1, and VOL2. "D03" is the code for a particular report, and it comes from the value of field RPT1. The value assigned to this parameter is the number of copies for the report, and its value comes from field CPY1. In this example, the number of copies becomes part of the dataset name, which is detected by a XEROX 120K off-line tape driven printer. In a typical print file, the D0 statement could be:

/REPORTS DD SYSOUT=A,COPY=5&003.

The numerical value from CPY1 would be substituted for "003". Note that no default values are coded on the PROC statement. There are three reasons for this: (1) by not coding default values, there can be no confusion as to the correct values for parameters (defaults versus values in the Monitor); (2) instead of frequently changing the
JCL (which is difficult because access to the PHOCLIB is restricted) it is much easier to change the appropriate fields in the Monitor; and (3) If one of the PROCs were erroneously executed, it would fail with a JCL error, thus precluding accidental execution.

SAS program B632SDLY is shown in Figure 3. This program runs each day as the last step of the "daily processing". The program reads the RAMDATE (processing date) and the password (from a RACF protected dataset). The DUE date on each Monitor record is then checked to determine if that job is due to run. If so, a jobcard is generated. If the output is to be routed to a remote printer, a ROUTE card is generated. If one or more tapes will be used, a SETUP card is generated. A few jobs do not follow the rules for tapes and the SETUP card; special SETUP cards are generated for these jobs according to IF statements. Next, the CPY and RMT fields are considered and the parameters for the EXEC card are strung together. After the EXEC card has been generated, another four lines of JCL are added to the jobstream.

The four lines are the UPDATE step which will be the last step of the job and will run only if all of the previous steps completed successfully (i.e., all steps return condition code zero). If this step runs, it executes SAS program B632SBU (Figure 4) which updates the Monitor's DUE date field for this job. The JOB code is passed into the program via the SYSIN option. When the matching Monitor record is found, the DUE field is updated according to the value in the PHED field. A daily job is updated as RAMDATE+1; weekly or bi-weekly jobs are updated as DUE+7 or DUE+14 respectively. RAMDATE is used for "daily" instead of DUE because "daily processing" does not run on weekends and holidays; the new DUE date would fall behind the RAMDATE, falsely indicating that the job was due to run (or had apparently ABENDED) even though it had probably run successfully. For monthly jobs, the new DUE date is usually the same day in the next month. This update is accomplished by breaking the date into month, day, and year. The month is increased by one. If the month becomes thirteen, it is reset to one and the year is increased by one. If the day part of the DUE date is 28 through 31, it can be assumed that the job is supposed to run on the last day of the month. For such jobs, the DUE date is advanced to the first day of the second month following; then the date is converted to SAS date notation and one is subtracted from the date. When this new date value is converted to MMDDYY format, SAS determines which day (number) is correct (SAS also takes leap year considerations). Similar computations are performed for quarterly, semi-annual, and annual jobs. The old and new DUE dates are shown in the SAS log.

The user may wish to change some of the values on the Monitor; however, some changes would require a corresponding change in the JCL. Since the user does not have access to the JCL, any such changes to the Monitor (as well as the JCL) must be done by the authorized software maintenance person. This has the added benefit of eliminating "finger pointing" when something goes awry, a la "too many cooks spoil the broth". User maintenance is restricted to the following fields: DUE, FREQ, NAME, CPY1 through CPY4, and RMT. Software maintenance personnel may change any field.

There are two different maintenance programs: one for ITY terminals, shown in Figure 5, and another for full-screen (3270-type) terminals, shown in Figure 6. The first program (B632SMUP) is a cross between interactive and batch update techniques — each change is validated interactively, but the change transactions are applied in a batch after all of the transactions have been entered and validated. Each transaction contains three elements: a JOB code, a field-name, and a new value for that field. Only one field may be changed per transaction. If a transaction fails a validation test, the appropriate diagnostic message is issued, the transaction is ignored, and the user may either re-enter the transaction or continue with other transactions. A valid transaction causes an IF statement to be generated. This statement is written to a file for a later step, which uses %INCLUDE to bring the IFs into the program where they can change a copy of the Monitor. The IF statements have the form:

IF JOB = 'job-code'
THEN field-name = new-value

After the changes have been processed,
the user has the option to abort, in which case none of the changes will be saved. If the abort option is not taken, the changes are saved by cycling through a PROC DATASETS / ACE operation. If the user realizes too late that he shouldn't have proceeded, there are four generations of the Monitor within the database, available for easy restoration by the software maintenance person. The full-screen version (F632PSFSE) is NOT transaction oriented. It uses SAS/FSP FSEDIT to permit changes to a copy of the Monitor. On the user's screen, certain fields are "protected fields", i.e. SAS/FSP will not permit the user to change the values in those fields. After exiting FSEDIT, the program subjects every record to validation (since it is not readily apparent which records were changed and which were not). Records that have errors are saved in SAS dataset ERRORS, along with text fields which contain the appropriate diagnostic messages. The program then enters FSBUI/SFSE to permit the user to view dataset ERRORS. If ERRORS is empty, then all of the changes were acceptable. As before, the user has the option to proceed or abort, with the same consequences.

**ACKNOWLEDGEMENT**

This paper is based on work performed under Contract Number AID/OTH-C-1055 between the Agency for International Development and Pinkerton Computer Consultants Inc.

**PROCESSING**

The daily processing program (B632XLY, Figure 3) is normally run as the last step of a batch Job. A CLIST, Figure 7, can also run B632XLY. If the positional parameter "XSUB" is specified, no jobs are submitted but the resultant listing shows which jobs would have been submitted. Depending upon when this is done, the listing could indicate which jobs are due to run, or which jobs should have run but apparently did not complete (had the jobs completed successfully, the last step of each job would have updated the Monitor and the job would no longer be listed as due to run).

**SUMMARY**

The advantages of this system are apparent in three areas. For the operations staff, there is no opportunity for a scheduling error since the jobs actually schedule themselves. For the software maintenance staff, JCL changes are reduced because those elements that change frequently are consolidated in the Monitor file. Additionally, it is easier to find which job produces which report by consulting the Monitor rather than by searching through documentation or through JCL libraries. For the user, there is the advantage of being able to make some operational changes by themselves (translation: responsiveness and independence); safeguards exist to prevent the user from inadvertently damaging any element of the system.

In other words, this approach improves reliability, ease of modification, and organization.
The following information is required:

jobcode: AB1 name: STAFF PATTERN
frequency: K due date: DEC 31, 1982 (enter ddmonyy ex: 05JUN82)

The following information is optional:

remote #: cpny1: 44 cpny2: cpny3: cpny4:

device information may be changed only by the DM analyst:

rpt1: D03 rpt2: rpt3: rpt4:
vol1: 001063 vol2: SCRCH vol3:

Figure 1.

//0632AB1 PROC D03=,VOL1=,VOL2=
//0632FIX5 EXEC PGM=0632FIX5 ...
//0632D10 EXEC PGM=0632DF00,PARM=R
/*003 ORGANIZATIONAL SUMMARY D501 TO63 Z 38 ...
(SOME JCL OMITTED DUE TO SPACE LIMITATIONS) ...
/*REPORTS DO USN=0632.TEMP,AA&D03,06,DISP=(NEW,PASS,DELETE),
   DCB=(RECFM=FBA,LMCOL=133,BLKSIZE=3990),
   SPACE=(TPK(120,90)),UNIT=TEM#
/*XEROX EXEC PGM=XEROX XEROX STEP, 42(?) COPIES OF SUMMARY PATTERN ...
/*SYSPRINT USN=J,(0034),DEST=LOCAL
/*PRINT01 USN=0632.TEMP,AA&D03,06,DISP=(OLD,PASS)
/*PTFORM EXEC PGM=PTFORM MICROFICHE STEP, 12 COPIES OF SUMMARY PNLN ...
/*PTFINPNT USN=0632.TEMP,AA&D03,06,DISP=(OLD,DELETE,DELETE)
/*EZFOUTPT USN=0632,FISP,DISP=(,KEEP,DELETE),UNIT=TAPE9,
/*EZFSTATS USN=SYSOUT=A,DCB=B,LKSIZE=2600
/*EZFPAPERS USN=DO13,PROC,CTRLCARD(0632AB1),DISP=SRH

FIGURE 2.

405
OPTIONS PS=60 LS=131
DATA SUBMITS = B632$DLY; SUBMITS ALL JOBS DUE TO RUN TODAY;
/* DDNAME 
USE DISP 
B632DATE RAMPDATE FROM LISTCAT SHR 
DB RAMP$ MONITOR DATABASE SHR 
INWRJ JCL FILE FOR JOB SUBMISSION OLD 
SUBLIST PRINT FILE FOR REPORT OF SUBMISSIONS OLD 
XTRA PARM INFO SHR */
LENGTH PARMX $ 4S;
RETAIN VC1 ',VOL1= VC2 ',VOL2= VC3 ',VOL3= RAMPDATE PARM LP;
ARRAY HPTS $3 HPT1-HPT4; ARRAY CPYS CPY1-CPY4;
ARRAY VOLS $6 VOL1-VOL3; ARRAY VC $6 VC1-VC3;
IF _N_=1 THEN DD/* ONLY READ RAMPDATE & PARM ONCE */
INFILE B632DATE; INPUT # 827 RAMPDATE WORDDATE.;
INFILE XTRA; INPUT PARM $1 LP=LENGTH(PARM);
END;
SET DB.MONITOR: FILE INTROH NOPRINT CLOSE=FREQ;
IF DUE <= RAMPDATE AND DUE = . THEN JOB
PUT '/B632' JOB $3. JOB (RPBC4,15.,99.,0.,0.) RAMP$CLASS=G,'/'
/* NOTIFY=B632HP,USER=B632HP,PASSWORD= PARM S VARYING. LP ;
IF HMT=., THEN DD/*
IF HAT<10 THEN PUT /*ROUTE PRINT HAT* HAT 1.1
ELSE PUT /*ROUTE PRINT HAT* HAT 2.1;
END;
IF VOL1=., THEN PUT /*SETUP+9 (VOL1-VOL3) ($6.,.);*
IF JOB='Z01' THEN PUT /*SETUP CATALOGOLED TAPE FOR B6325011;
IF JOB='Z0X' OR JOB='A51' OR JOB='A52' OR JOB='A61' OR JOB='A62' THEN PUT /*SETUP CATALOGOLED TAPE RUN B632.B1B11;
IF JOB='Z21' THEN PUT /SETUP WILL CALL FOR ' H632.XML FROM JOB B632Z011 / /*SETUP RUN AER JOB B632Z011
IF SUM(CPYS-CPY4)='. THEN DD OVER RPTS;
IF HPTS=.' AND CPYS=. THEN JOB;
IF _N_=1 THEN PARMX=,'/HPT$11=TRIM(LEFT(PUT(CPYS,Z2.)));"
ELSE PARMX=TRIM(PARMX)111,'/HPT$11=TRIM(LEFT(PUT(CPYS,Z2.)));"
END;
END;
END;
DO OVER VOLS;
IF PARMX=.' AND VOLS=.' THEN PARMX=VC11VOLS;
ELSE IF PARMX=.' AND VOLS='.' THEN PARMX=TRIM(PARMX)111VC11VOLS:
END;
PUT '/B632 EXEC B632 JOB $3. PARMX $48.1
PUT '/B63250AT EXEC SAS,OPTS=/*NOSOLUTION SYSPARM= */
/*NOJOB $3. */,'/COND=(G,NE)/*
/*SYSIN DD DSN=0.013.PRD.SORLIB(B632$RUN),DISP=SHR/ /
/*DB DD DSN=B632.MONITOR,DISP=OLD/ /
/*B632DATE DD DSN=B632.DATE,DISP=SHR/ */*/;
END;
ELSE DELETE;
DATA _NULL_;
SET SUBMITS: FILE SUBLIST PRINT HEADER=HUR;
PUT $5 JOB $3. #9 DUE WORDDATE12. #24 FREQ $1. #27 NAME; 
RETURN;
HUR: PUT 'JOBS DUE/SUBMITTED FOR RAMPDATE ' RAMPDATE WORDDATE12. /;
PUT $5 'JOB' $6 'DUE DATE' #22 'FREQ REPORT NAME' /;
RETURN;

FIGURE 3.
OPTIONS GEN=1; /* B6326U RUN UPDATE MONITOR AT END OF EACH JOB */
DATA B6,MONITOR; SET B6,MONITOR;
DUMP CHARDATE U M Y OLD_DUE RAMPDATE LOOP;
RETAIN RAMPDATE;
IF _N_=1 THEN DO; /* GET RAMPDATE ONCE */
   INFILE B632DATE; INPUT #9 #27 RAMPDATE WM(DDDYY6.); 
END;
IF JOB=SYSPARM() THEN DO:
   OLD_DUE=DUE;
   IF FREQ='X' THEN DUE=1;
   IF FREQ='D' THEN DUE=RAMPDATE+1;
   IF FREQ='W' THEN DUE=DUE+7;
   IF FREQ='M' THEN DO;
      CHARDATE=PUT(DUE,WM(MMDDYY6.));
      D=SUBSTR(CHARDATE,3,2); Y=YEAR(DUE); M=MONTH(DUE)+1;
      IF M=13 THEN DO;
         M=1;
         Y=Y+1;
      END;
      DUE=MDY(M,D,Y);
      IF DUE=. OR D>=28 THEN DO;
         D=I;
         M=M+1;
         IF M=13 THEN DO;
            M=1;
            Y=Y+1;
         END;
      DUE=MDY(M,D,Y)-1;
   END;
   IF FREQ='O' THEN DO;
      M=MONTH(DUE); Y=YEAR(DUE); D=I;
      IF M>12 THEN DO;
         M=12;
         Y=Y+1;
      END;
      DUE=MDY(M,D,Y)-1;
   END;
   IF FREQ='S' THEN DO;
      M=MONTH(DUE); Y=YEAR(DUE); D=I;
      IF M>12 THEN DO;
         M=M-12;
         Y=Y+1;
      END;
      DUE=MDY(M,D,Y)-1;
   END;
   IF FREQ='A' THEN DO;
      CHARDATE=PUT(DUE,WM(MMDDYY6.));
      SUBSTR(CHARDATE,5,2)=SUBSTR(CHARDATE,6,2)+1;
      DUE=PUT(CHARDATE,WM(MMDDYY6.));
   END;
   PUT / OLD_DUE= YWMMDD8. +2 DUE= YWMMDD8. +2 FREQ= +2 JOB= +2;
   NAME RAMPDATE= YWMMDD8. //;
END;

**FIGURE 4.**
DATA _NULL_; INFILF TERMINAL X=TODAY();
PUT '8632SMUP -- RAMPS MONITOR UPDATE';
PUT 'TODAY IS \ X YYYYMMDD, (' X 4. ')';
PUT 'DO YOU NEED INSTRUCTIONS? (TYPE Y FOR YES, N FOR NO)';
INPUT ANS $;
IF ANS='Y' THEN DO;
PUT '/*TRANSACTION FORMATT JOBCODE FIELD-NAME NEW-VALUE*/';
PUT 'WHEN CHANGING THE NAME OF A REPORT, THE NEW NAME MUST BE/'
ENCLOSED IN 'SINGLE QUOTES AND NOT MORE THAN 40 CHARACTERS'/'
-- EXAMPLE: *AB NAME *STAFFING PATTERN*/';
PUT 'DATE-DATE MUST BE IN YYYYMM FORMAT -- TO DELETE A DATE, USE'/'
'999999' /'ON A PERIOD (.) FOR THE NEW VALUE';
PUT 'ONLY 1 FIELD MAY BE CHANGED PER TRANSACTION';
PUT 'EXAMPLE: */AAI FREQ D*/AAI CPY1 TO*/AAI CPY2*/';
END;
STOP;
DATA; INFILF TERMINAL MISSOVER; DROP TEST 1: LENGTH VAR $8 VAL $42;
PUT 'ENTER JOBCODE (OR END) FIELD-NAME NEW-VALUE';
INPUT JOBCODE $1-3 VAR $5 8 CHAR; @4 TEST CHAR.
IF JOBCODE='END' & TEST = ' ' THEN STOP;
IF JOBCODE=' ' OR ((VAR=' ' OR VAL=' ') & JOBCODE='='='END') THEN DO;
PUT 'ERROR: MISSING JOBCODE OR FIELD-NAME OR NEW-VALUE -- '/
TRANSACTION REJECTED';
DELETE;
END;
IF VERIFY(JOBCODE, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')'=O
OR LENGTH(JOBCODE)<3 OR TEST=' ' THEN DO;
PUT 'ERROR: JOBCODE MUST BE 3 ALPHANUMERIC CHARACTERS -- '/
TRANSACTION REJECTED';
DELETE;
END;
IF JOBCODE=' ' &
(VERIFY(SUBSTR(JOBCODE,1,1), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')'=O
OR VERIFY(SUBSTR(JOBCODE,2,1), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')'=O
OR VERIFY(SUBSTR(JOBCODE,3,1), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')'=O
THEN DO;
PUT 'ERROR: INVALID JOB CODE -- TRANSACTION REJECTED';
DELETE;
END;
IF VAR='DUE' & VAR='FREQ' & VAR='NAME' & VAR='CPY1' &
VAR='CPY2' & VAR='CPY3' & VAR='CPY4' & VAR='RMT' THEN DO;
PUT 'ERROR: INVALID FIELD NAME -- VALID NAMES ARE';/
'DUE, FREQ, NAME, CPY1 THRU CPY4, AND RMT'/
TRANSACTION REJECTED';
DELETE;
END;
END;
IF VERIFY(TRIM(VAL), '0')=O THEN VAL='.';
L=LENGTH(TRIM(VAL));
IF VAR='CPY' THEN DO;
IF (VERIFY(TRIM(VAL), '01234567890')'=O & VAL=' ') OR L>2 THEN DO;
PUT 'ERROR: INVALID VALUE FOR VAR SCHARS. VAL SYVARING. L';/
'98 VALUE MUST BE A NUMBER BETWEEN 1 AND 99 -- '/
TRANSACTION REJECTED';
DELETE;
END;
END;
FIGURE 5 (PART 1 OF 2).
IF VAR='FcltO' THEN DO;
   IF INDEXE(VAL, 'XLMEMSCE') =0 OR L=1 THEN DO;
      PUT 'ERROR: INVALID VALUE FOR FREQ' / # $ 'VALID VALUES ARE' /
         ' ' X a b c d e f g h i j k l m n o p -- TRANSACTION REJECTED';
   DELETE;
END;
ELSE VAL=TRIM(VAL)!!!'/';
END;
IF VAR='DU' THEN DO:
 IF VAL='999999' THEN VAL='.' INPUT (VAL, YMMDD6.);
 IF _ERROR_ OR (L=6 & VAL='.' ) OR VAL='....' THEN DO;
    _ERROR_=1;
 IF SUBSTR(VAL,1,1)=' ' THEN PUT 'ERROR: THERE MAY BE ONLY 1' /
       ' BLANK BETWEEN "DUE" AND". / #$ "THE NEW DATE VALUE" /
       " -- TRANSACTION REJECTED';
 ELSE PUT 'ERROR: INVALID DATE / VAL $VARYING. L / # $ "DATE" /
       MUST BE IN YMMDD FORMAT, OR 999999 -- TRANSACTION REJECTED';
 DELETE;
END;
END;
END;
ELSE PUT=PUT(.4.);
IF VAR='NAME' AND (SUBSTR(VAL,1,1)='.' OR
SUBSTR(VAL,1,1)='"') THEN DO;
 PUT 'ERROR: VALUE SHOULD BE IN SINGLE QUOTES, OR IS MORE THAN 
# $ 40 CHARACTERS LONG -- TRANSACTION REJECTED';
 DELETE;
END;
IF VAR='RMT' & (L<3 OR VERIFY(TRIM(VAL),'.14$'))=0 THEN DO;
 PUT 'ERROR: VALUE FOR RMT MUST BE 1,4,5 OR A PERIOD (.);' /
# $ "TRANSACTION REJECTED";
 DELETE;
END;
END;
RUN;
DATA _NULL_; SET TEMP NOPRINT; L=LENGTH(TRIM(VAL));
 PUT @5 IF JOb='3' 'JOBCODE $3. ' THEN 'VAR=VAL $VARYING. L ' /;
 PROC PRINT; FORMAT _ YMMDD6.; VAR JOBCODE VAR VAL _ ; RUN;
 OPTIONS NOTES;
DATA DB.NEW; SET DB.MONITOR; %INCLUDE TEMP; RUN;
DATA _NULL_; FILE TERMINAL;
   PUT 'IF CHANGES ARE ACCEPTABLE, RESPOND GO TO CHANGE THE' /
      ' MONITOR FILE -- OTHERWISE, RESPOND ABORT TO ABORT';
   INPUT ANSWER = 1; IF ANSWER='GO' THEN STOP;
   ELSE IF ANSWER='ABORT' THEN DO;
      PUT 'ABORT ACCEPTED -- SESSION WILL END VIA ABORT "ERROR"';
      ABORT ABEND;
   END;
   ELSE PUT 'ERROR: INVALID RESPONSE' / ;
 PROC DATASETS CNAMEx=DATA;
 AGE NEW MONITOR MONITOR1 MONITOR2 MONITOR3 MONITOR4;

FIGURE 5 (PART 2 OF 2).
* B632FSE MONITOR FULL SCREEN EDIT  ROSS Z MEHLMAN 1
* PINEGTON COMPUTER CONSULTANTS INC. W/0 3065 6/21/82 1

OPTIONS GEN=11
DATA DB.New SET DB.MONITOR
PROC FSEDIT DATA=DB.New SCREEN=SCRLIB.B632ANAL OPTION=11
PROC SORTI DATA=DB.New BY JOB;
DATA ERRORS SET DE.NEW DROP X CPYX RMTX;
LENGTH ERR=ERR8 $40 CPYX RMTX $2;
ARRAY ERR (X) ERR=ERR81 ARRAY CPY CPY=CPY4;
X=11
IF JOB=''J16' 
& (VERIFY(SUBSTR(JOB,1,1),'ABCDEFGHIJKLMNOPQRSTUVWXYZ')=0 OR 
VERIFY(SUBSTR(JOB,2,1),'ABCDEFGHIJKLMNOPQRSTUVWXYZ')=0) 
& (VERIFY(SUBSTR(JOB,3,1),'0123456789')=0) THEN DO;
ERR='INVALID JOB CODE' ; 
X=X+11
END;
DO OVER CPY1;
CPYX=LEFT(PUT(CPY,2,1))
IF (VERIFY(TRIM(CPYX),''0123456789'')=0) 
& (VERIFY(TRIM(CPYX),''ABCDEFGHIJKLMNOPQRSTUVWXYZ'')=0) 
& (VERIFY(TRIM(CPYX),''0123456789'')=0) THEN DO;
ERR='INVALID VALUE FOR CPY'
X=X+11
END;
IF INDEX(FREQ,'CDWMOSA')=0 OR LENGTH(FREQ)=1 THEN DO;
ERR='INVALID VALUE FOR FREQ' ;
X=X+11
END;
IF FREQ=''X'' AND DUE='' THEN DO;
ERR='DUE DATE MUST BE SPECIFIED' ;
X=X+11
END;
RMTX=LEFT(PUT(RMT,2,1))
IF (VERIFY(TRIM(RMTX),''145'')=0) THEN DO;
ERR='VALUE FOR RMT MUST BE 1,4,5 OR A PERIOD' ;
X=X+11
END;
IF X>1 THEN OUTPUT;
RUN.
PROC FSBROWSE DATA=ERRORS OPTION=11;
DATA _NULL_ INFILE TERMINAL;
PUT 'IF CHANGES ARE ACCEPTABLE, RESPOND GO TO CHANGE THE' ;
'MONITOR FILE -- OTHERWISE, RESPOND ABORT TO ABORT.';
INPUT ANSWER $ ;
IF ANSWER=''Y'' THEN STOP;
ELSE IF ANSWER=''ABORT'' THEN DO;
PUT 'ABORT ACCEPTED -- SESSION WILL END VIA ABORT "ERROR"';
ABORT ABORT;
END;
ELSE PUT 'ERROR: INVALID RESPONSE' ;

PROC DATASETS DONAME=DB;
AGE NEW MONITOR MONITOR1 MONITOR2 MONITOR3 MONITOR4

FIGURE 6.
PROCEDURE NOSUB DPRINT(05020D08)

WHITE-RAILS DAILY ASYSTIME &SYSUATE

/*
* XB632$DLY SUBMIT ALL RAMPS DAILY JOBS DUE TODAY (RAMPDATE) */
/* IF PARAM 'NOSUB' IS SPECIFIED, JOBS WILL NOT BE SUBMITTED */
/* IF DPRINT* IS USED, SUBLIST WILL BE WRITTEN TO THE */
/*oram PRINT */
/*========================================================================*/
ENRON DO

IF &LASTCC NE 12 AND &LASTCC NE 8 AND &LASTCC NE 976 THEN -
WHITE ERROR CONDITION &LASTCC PROCESSING CONTINUES
RETURN
END

CONTROL NOSMSG
IF &SYSPROC=INIT THEN PROFILE PREFIX(B632MP) LIST
IF &SYSPROC=INIT THEN SET DPRINT=STR(*)
FREE ATTLIST(VBA HDR)
DELETE 'B632.SUBLIST'
CONTROL MSG
ATTR VBA LRECL(137) BLKSIZE(19069) RECFM(V B A) USORC(PS)
/* B632DATE ALLOCATED USING(VBA) SPACE(1) TRACKS VOL=ALIVOL */
ALLOC FCB(B632DATE) DA('B632.DATE') OLD REUSE
LISTCAT LNT(B632.DATE) VOL OFILE(R632DATE)
/*========================================================================*/
IF &NOSUB=NOSUB THEN ALLOC F(INTIDR) DUMMY REUSE
ELSE ALLOC F(INTIDR) DA('B632.SUBJCL') OLD REUSE
IF &SYSPROC NE INIT THEN ALLOC F(DOB) DA('B632.MONITOR') SHR REUSE
/* FILE DB ALLOCATED IN JCL FOR BATCH RUNS */
IF &STR(DPRINT)=STR(*) THEN ALLOC F(SUBLIST) DA(*) SHR REUSE
ELSE ALLOC F(SUBLIST) DA('B632.SUBLIST') NEW REUSE USING(VBA) -
SPACE(1) TRACKS
ALLOC F(D) DUMMY REUSE /* PROGRAM LOG -- DA(*) FOR DEBUGGING */
ALLOC F(F) DA('0013.PROD.SORLIB(B632DLY)') SHR REUSE
ALLOC F(DATA) DA('B632.X807') SHR REUSE
/* B632DATE STILL ALLOCATED FROM ABOVE */
SAS OPTIONS='SYSIN=I LOG=L')
/*========================================================================*/
FREE F(DU)

IF &NOSUB NOSUB THEN DO
CONTROL NOSMSG
SUBMIT 'B632.SUBJCL'
CONTROL NOSMSG
ATTR HDR LRECL(80) BLKSIZE(3120) RECFM(F B A) USORC(PS)
ALLOC F(SYSIN) DUMMY REUSE
ALLOC F(SYSPRINT) DA(*) REUSE
ALLOC F(SYSUT1) DUMMY USING(HDR) REUSE
ALLOC F(SYSUT2) DA('B632.5SUBJCL') OLD REUSE
CALL 'SYSLINKLIB(IEBOBENER)' /* THE ABOVE OPERATION CLEARS-OUT THE JCL FILE */
END
END
CONTROL NOSMSG
IF &STR(DPRINT) NE &STR(*) THEN DO
DSPRINT 'B632.SUBLIST' ADSPRINT NOUNOM CCHAR
DELETE 'B632.SUBLIST'
END

FIGURE 7.