

**Paper 184-2009****Automating your SAS® jobs**

Paul Anderson, Alere, Irvine, CA

**Abstract**

This paper discusses automating formerly ad hoc jobs that have become recurring daily, weekly, or monthly jobs. Topics covered include naming conventions that facilitate processing, assigning dates, creating historic processing logs, e-mailing results, and scheduling and triggering the processing.

**Introduction**

I had been a mainframe programmer for 20 years and now I was working with this system with very little structure. No real system in place. Only the programmer knew where the programs and files were located, and who to notify that the report or output files were ready. No history log of the job being run, no program logs saved from one run to the next run. The program code was saved in the same folder that the input and output files were also stored. When we had our first employee turnover, we had to determine what ad hoc jobs the employee was running, what files were being created, and who was being notified. When were the files created? Was there a schedule? Where was the code stored? In many cases it was on the employees 'C' drive. We needed some structure. We needed an auditable system.

**First Tasks**

Set up an Informatics email address. All incoming and outgoing emails for production processing would be through the Informatics email address. This way if we have employee turnover, we would still receive the files needed, and we would also have a history of notifications or files delivered. And we would not have to notify clients that an employee was lost.

Move all programs to one location. A location on a server where all members of the group could access and run the programs.

Save all program log files with the program name and date of processing embedded in the file name to one location.

Write one line of information to a job run log that indicates the date, time, program, and submitter of the program.

Backup the program folder every night naming the backup folder yyyyymmdd.

### **Folder and file naming specifications**

All of our original input data from our customers are loaded into ZCLIENTDATA\XYZCo\yyyymm.

All of the temporary files created during the job run write to Tclientdata\XYZCo\yyyymm.

All of the permanent files created during the job run write to Pclientdata\XYZCo\yyyymm.

And our output files for the client are written to XYZCo\.

Our output file structure might look like 'XYZCo\COPD\ Referral yyyymmdd'.

The yyyymmdd is important so that we can sort the files in the folder.

### **Program redesign**

Now that we have protected our programs, we need to develop a system to automatically assign dates, identify the files to be processed, and uniquely identify the files created.

We first developed a macro that is executed at the beginning of every job to create the current date as yyyymmdd and current month as yyyymm. The current date is used when creating the log and lst files. And also as a date processed stamp added to the job log.

We may also embed this date into any files created. By embedding this date we can sort the files in the folder and process the most recent or X most recent files in a folder.

The current month date is used to identify a folder that may hold the current months data and/or where to write the current months data.

Additionally we added other code using the intnx function to identify date ranges to be processed.

### **Emailing**

When the program finishes an email is sent to the Informatics email address and other interested persons.

This email contains a list of files processed, files created, control totals, and any other pertinent information that the user may deem valuable to assure them the data was processed correctly.

### **Scheduling**

Using the Microsoft scheduler we have set up the same job to execute 4 times a day and three times during the evening. The job consists of day checks, time checks, and file exists checks to control the timing of our jobs. Within many of our jobs the last step will

create an empty file called a trigger file. The schedule then checks to see if the trigger file exists, if it does, the next job is submitted.

### Benefits

No longer have to change code before each run.  
 Forced Standards.  
 Only QA once  
 Can set up on scheduler  
 Not effected by employee turnover

### Problems

Employee worries about losing control  
 Management must restructure staff

Included below

Code example.

1. Program code
2. Macro date code
3. Macro to read all files in folder
4. Macro to write to joblog
5. Email code
6. Macro trigger code

### Program Example

```

/*=====
/ Organization : Alere Medical, Inc.
/ Purpose      : Identification of Diabetes
/ SubMacros    :
/
/ Notes       :
/
/ Usage       : choose either calendar year reporting period or non calendar
/              year reporting period depending on the plan.
/=====
/=====
/ This is proprietary and confidential information. Not to be disclosed.
/=====*/

/*delete trigger file that fileexist check before submitting job*/
X del 'm:\Production\Triggers\XXX_Diabetes_Referrals.txt';

options mautosource sasautos=("m:\paul\mymacros") noxwait noxsync formdlm=' ';
run;

%let fdate=%sysfunc(today(),yymmddn8.);
%let mdate=%sysfunc(today(),yymmnn6.);

%fgen(m:Zclientdata\XXX\DIAB,1,xfile1);

```

```

%put &fdate &mdate &pgm;

proc printto log="M:\PRODUCTION\Monthly\LOGS\XXX_Diabetes_Referral_&fdate..log" new;
proc printto print="M:\PRODUCTION\Monthly\LOGS\XXX_Diabetes_Referral_&fdate..lst" new;

*****;
***Get the previous members referred *****;
*****;
%drivexh(M:\XXX\Monthly output\West\DIAB,diab_prev);
run;

proc sort data=diab_prev(keep=hhtempl_member_id) nodupkey; by hhtempl_member_id;
run;
proc sql;
  select count(hhtempl_member_id) into :diabdups
  from diab_prev;
quit;
%put &diabdups;

*****;

data diabetes;
infile "&xfile1" lrecl=1000 trunccover dsd;
input

  input code

;
run;
proc sql;
create table West_Diabetes as
select *,
"REFERRALALERE" AS FILLER,
"High" as Acuity_flag,
" " as HHTEMPL_MARKET_ID format=$5.,
" " as HHTEMPL_Market_Name format=$25.
FROM Diabetes
where substr(HHTEMPL_BASE_MKT,1,3) in (List of codes)
and HHTEMPL_CONSOL_MKT in (List of codes)

order by HHTEMPL_MEMBER_ID;
QUIT;
proc sort data=diab_prev(keep=hhtempl_member_id) nodupkey; by hhtempl_member_id; run;
proc sort data=West_diabetes nodupkey; by hhtempl_member_id; run;

/*Remove previously sent members from new list*/

data a_and_b a_not_b b_not_a lost;
merge West_diabetes(in=a) diab_prev(in=b);
by hhtempl_member_id;
if a and b then output a_and_b;
else if a and not b then output a_not_b;
else if b and not a then output b_not_a;
else output lost;
run;

```

```
proc sql noprint; create table nrefcnt as select count(*) into: n_referrals from a_not_b;
quit;
proc sql noprint; create table prefcnt as select count(*) into: p_referrals from a_and_b;
quit;

%put &p_referrals &n_referrals;

PROC EXPORT DATA= work.a_not_b
      OUTFILE= "M:\XXX\Monthly output\West\DIAB\XXX West_ALR_DIA_REFL_&fdate..csv"
      DBMS=CSV REPLACE;
RUN;

PROC EXPORT DATA= WORK.a_and_b
      OUTFILE= "C:\XXX West_ALR_DIA_REFL_Duplicates_&fdate..csv"
      DBMS=CSV REPLACE;
RUN;

data _null_;
x=sleep(2);
run;

filename outbox email;
  data _null_;
    file outbox
      to=('panderson@alere.com')

      subject="XXX West_ALR_DIA_REFL_&fdate"
      attach=("M:\XXX\Monthly output\West\DIAB\XXX West_ALR_DIA_REFL_&fdate..csv" );
    put "Alere Teammate,";
    put "Attached is the referral file.";
    put " ";
    put "The program is &pgm";
    put " ";
    put "The input file is &ifile1";
    put " ";
    put "Number of new referrals &n_referrals";
    put " ";
    put "Number of previous referrals &p_referrals";
    put " ";
    put "Please let me know if you have any questions.";
    put " ";
    put "Thank You,";
    put "Paul Anderson";
    put "Senior Healthcare Analyst";
    put " ";
    put "ph: 949-735-6207";
    put "email: panderson@alere.com";
run;

data _null_;
x=sleep(2);
run;
```

```

%macro log_write;

filename out "M:\Production\LOGS\Log.txt";
data _null_;

    today =today();
    file out mod;
    put today yymmddn8. ',' "&stime" ',' "&sysuserid" ',' "XXX_Diabetes_Referral";
run;
%mend log_write;

%log_write;

data _null_;
x=sleep(2);
run;

%macro write_trigger(fname);
/*creates empty file that the schedule does a fileexist against to submit the next job*/
filename out "M:\PRODUCTION\Triggers\&fname.txt";
data _null_;
    file out;
    put "&sysdate" ',' "&stime" ',' "&sysuserid" ',' "&fname";
run;
%mend write_trigger;

%write_trigger(XXX_Diabetes_Enrollment_Tracking);

data _null_;
x=sleep(2);
run;

*** end of program ****

```

## Macro fgen

\*\*\*This macro gets the &gen file created in a folder.\*\*\*

```

%macro fgen(fname,gen,mcn);
*This macro will find the generation data set;

filename DIRLIST pipe %unquote(%str('%dir "&fname" %'));

data dirlist;
    length buffer $256 filenm $65;
    infile dirlist length=reclen;
    input buffer $varying256. reclen;
    if index(buffer,'xls') or index(buffer,'txt');
    filenm=substr(buffer,40,65);

    startd=indexc(filenm,'0123456789') ;
    endd=index(filenm,')-startd;
    date=input(substr(filenm,startd,endd),mmdyy10.);

```

```

run;
proc sort data=dirlist; by descending filenm; run;

data _null_;
  set dirlist(firstobs=&gen obs=&gen);

  call symput("&mcn",filenm);
run;

%put &&mcn;

%mend fgen;

```

## Macro Drivexh

/\* reads all files in a folder \*/

```

%macro drivexh(dir,dsname);

  %let filrf=mydir;
  %let rc=%sysfunc(filename(filrf,&dir));
  %let did=%sysfunc(dopen(&filrf));
  %let memcnt=%sysfunc(dnum(&did));
  %put NUMBER OF FILES IN FOLDER IS &MEMCNT;
  %if &MEMCNT=0 %then %abort abend 333;

  %do i = 1 %to &memcnt;
    %let name=%qscan(%qsysfunc(dread(&did,&i)), -1,.);
    %if %qupcase(%qsysfunc(dread(&did,&i))) ne %qupcase(&ext) %then %do;
    %if (%superq(ext) ne and %qupcase(&name) = %qupcase(&ext)) or
      (%superq(ext) = and %superq(name) ne) %then
      %do;
      PROC IMPORT OUT= xxx DATAFILE= "&dir\%qsysfunc(dread(&did,&i))" DBMS=CSV
REPLACE;
      GETNAMES=YES; DATAROW=2; RUN;
      %let ifile=%qsysfunc(dread(&did,&i));
      data xxx; set xxx; inputfile="&ifile"; run;

      %put %qsysfunc(dread(&did,&i));
      proc append base=&dsname data=xxx force; run;
      %end;
    %end;
  %end;
  %let rc=%sysfunc(dclose(&did));
%mend drivexh;

```

## Contact Information

Paul Anderson  
 170 West Avenida San Antonio  
 San Clemente, CA 92672  
 949 735 6207  
[PSA170@hotmail.com](mailto:PSA170@hotmail.com)

SAS is a registered trademark of SAS Institute, Inc

in the USA and other countries. ® indicates USA registration