

## SAS Automation and More

Frank Fan, Abbott Vascular, Santa Clara, CA

### ABSTRACT

SAS programmers/statistical analysts spend much time on repeating tasks, such as rerunning programs with different parameters and creating similar reports. Using different techniques, we automate the tasks to free the programmers from these tedious, manual burdens. These techniques include the Windows task scheduler, script/batch files, method to find a specific input from a folder, SAS tools like the email file engine, macros to identify error/warning messages in the SAS log file, etc. This paper will demonstrate how we use these tools to benefit the programmers.

### INTRODUCTION

The trend of SAS programmer's time spent on coding and maintenance should be reduced so that they can concentrate more on data analysis and data science fields (Smoak 2015). CDISC and macros are ways to move towards this goal. This paper explores some more specific methods to free programmers from these time-consuming tasks. These methods are as follows:

- Schedule the batch file with SAS programs wrapped in (to run at scheduled time)
- Subgroup analysis automation
- Sending out email from a SAS program
- Miscellaneous automation tips
  - Check error/warning in SAS log file
  - Find the latest file in a folder to use as input
  - Find the latest directory in a folder

### SCHEDULING THE BATCH FILE WITH SAS PROGRAMS WRAPPED IN

Scripting is a typical means of automation traditionally. Batch file is the version of the script file in Windows, which is a text file containing one or more DOS commands and saved with ".bat" as the file extension. When it is run, the commands are executed sequentially. We use it to run SAS programs in Windows. Here is an example.

```
start/w " " "C:\Program Files\SAS\SAS 9.1\sas.exe" -CONFIG "C:\Program Files\SAS\SAS 9.1\SASV9.CFG" -SYSIN "\\sclnas01\project\SHRDATA\ASE\Study1\spgm\PDListing.sas" -LOG "\\sclnas01\project\SHRDATA\ASE\Study1\slog\PDListing.log"

start/w " " "C:\Program Files\SAS\SAS 9.1\sas.exe" -CONFIG "C:\Program Files\SAS\SAS 9.1\SASV9.CFG" -SYSIN "\\sclnas01\project\SHRDATA\ASE\Study2\spgm\ad_ae_worksheet.sas" -LOG "\\sclnas01\project\SHRDATA\ASE\Study2\slog\AEReport.log"
```

When the batch file above runs, the first program "PDListing.sas" will be executed first, followed by "ad\_ae\_worksheet.sas" sequentially. The log files, as specified in "-LOG" part, are then saved accordingly. Note the above batch file assumes your SAS is installed in the "C:\Program Files\SAS\SAS 9.1" directory. The user may need to adjust the path to where the SAS execute file located. The Windows system first opens SAS.exe, then runs the SAS program under "SYSIN" from SAS.

Once a batch file is created, the user can run it immediately by double clicking it or scheduling it to run at a later time. The following instruction shows how to schedule a batch file in the Windows 7 and 10 systems. If one uses Unix/Linux, please refer to Michael B. Spring in the Reference section for details. Depending on the company's policy/setting, one may need to request local administrator's right to setup and execute scheduled tasks.

- Open "Task Scheduler" by clicking the "Start button", clicking "Control Panel", clicking "System and Security", clicking "Administrative Tools", and finally double-clicking "Task Scheduler". If you're prompted for an administrator password or confirmation, type the password or provide confirmation.
- Click the "Action" menu, followed by "Create Basic Task".

- Type a name for the task and an optional description, and then click “Next”.
- Do one of the following:
  - To select a schedule based on the calendar, click “Daily”, “Weekly”, “Monthly”, or “One time”; click “Next”; specify the schedule you want to use, and then click Next.
  - To select a schedule based on common recurring events, click “When the computer starts” or “When I log on”, and then click “Next”.
  - To select a schedule based on specific events, click “When a specific event is logged”; click “Next”, specify the event log and other information using the drop-down lists, and then click “Next”.
  - To schedule a program to start automatically, click “Start a program”, and then click “Next”.
- Click “Browse” to find the program you want to start, and then click “Next”.
- Click “Finish”.

We can run several programs in one click once the programs are put in a single batch file. For example, we usually wrap our ADS programs in a batch file for each study, then for each analysis, the expected ADS are generate by executing the batch file. Of course, the programs should be portable, i.e. use relative path, and keep the folder structure similar. Here is an example batch file that extracts data from the database. Note that the batch file is robust enough to deal with different SAS versions, or different locations the SAS system installed:

```
REM: A BATCH PROGRAM TO EXTRACT DATA FROM DATABASE
if exist "C:\Program Files\SASHome\SASFoundation\9.3\sas.exe" goto sasc93
if exist "C:\Program Files\SAS\SASFoundation\9.2\sas.exe" goto sasc92
if exist "E:\Program Files\SAS\SASFoundation\9.2\sas.exe" goto sase92
if exist "C:\Program Files\SAS\SAS 9.1\sas.exe" goto sasc91
:SASC93
set sas="C:\Program Files\SASHome\SASFoundation\9.3\sas.exe"
  -CONFIG "C:\Program Files\SASHome\SASFoundation\9.3\nls\u8\sasv9.cfg"
goto ENDCOND

:SASC92
set sas="C:\Program Files\SAS\SASFoundation\9.2\sas.exe"
  -CONFIG "C:\Program Files\SAS\SASFoundation\9.2\nls\u8\SASV9.CFG"
goto ENDCOND

:SASE92
set sas="E:\Program Files\SAS\SASFoundation\9.2\sas.exe"
  -CONFIG "E:\Program Files\SAS\SASFoundation\9.2\nls\u8\SASV9.CFG"
goto ENDCOND

:SASC91
set sas="C:\Program Files\SAS\SAS 9.1\sas.exe"
  -CONFIG "C:\Program Files\SAS\SAS 9.1\nls\en\SASV9.CFG"
:ENDCOND
```

```
REM: SPECIFY FILE/PROGRAM PATHS
set plog=-log "S:\SHRDATA\ASE\08-107 STUDY1\slog\"
set ppth=-sysin "S:\SHRDATA\ASE\08-107 STUDY1\spgm\extract\
REM: CREATE OUTPUT AND STORE SAS LOG/LST FILES TO SPECIFIED PLACES
%sas% %ppth%import.sas" %plog%
%sas% %ppth%STUDY1_dset_trim.sas" %plog%
```

```
REM: SPECIFY FILE/PROGRAM PATHS
set plog=-log "S:\SHRDATA\ASE\05-102 STUDY2\slog\"
set ppth=-sysin "S:\SHRDATA\ASE\05-102 STUDY2\spgm\extract\
REM: CREATE OUTPUT AND STORE SAS LOG/LST FILES TO SPECIFIED PLACES
%sas% %ppth%import.sas" %plog%
%sas% %ppth%STUDY2_dset_trim.sas" %plog%
```

REM: END OF BATCH PROGRAM

## SUBGROUP ANALYSIS AUTOMATION

With the idea of the batch file introduced above, we can run any of our programs, like ADS or TLG, automatically by simply creating the corresponding batch files. Here is a batch file example to demonstrate the subgroup analysis with SAS programs wrapped in a batch file.

```
X:

cd "X:\Clinical Affairs\BDM\ClinicalStudies\Integrated\Pooled
2015\Analysis\Programs"

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_baseline.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_mae.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_mae_secondary.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_30dMort.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_chf_hosp.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_procmort.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_procedure.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_mr2_matched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_mr2_unmatched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_nyha_matched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_nyha_unmatched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_lv_matched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_lv_unmatched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_qol_matched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_qol_unmatched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_meds_matched.sas -sysparm gender*Male*Female

start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -sysin
int_t_meds_imp.sas -sysparm gender*Male*Female
```

Note that instead of manually draft the batch file above, it can be generated by this SAS program:

```
data _null_;
  file "&batch_file..bat";
  put "X:";
  put 'cd "X:\Clinical Affairs\BDM\ClinicalStudies\study\Analysis\Programs"';
  %let cnt = %eval(%sysfunc(countc(&pgmLst,'*'))+1);
  %do i=1 %to &cnt;
    %let pgm = %scan(&pgmLst,&i,'*');
    /*next line is for windows 8 */
    put "start /wait /min C:\Progra~1\SASHome\SASFou~1\9.3\sas.exe -nosplash -
      sysin &pgm..sas -sysparm &var*&labelYes*&labelNo";

    /*next line is for windows 7 */
    /*put "start /wait /min C:\Progra~1\SAS\SASFou~1\9.2\sas.exe -nosplash -
      sysin &pgm..sas -sysparm &var*&labelYes*&labelNo";*/
  %end;
run;
```

## SENDING EMAILS IN SAS PROGRAMS

Batch files with scheduled tasks can automate our program execution and report generation. Sending emails from SAS programs further automates the whole process and sends the reports to the users directly on time. To realize that, we need to first modify the SAS configuration file, then follow the syntax as described below.

```
FILENAME fileref EMAIL <'address' ><email-options>
```

This filename statement must be defined with keyword “EMAIL” to specify the device type, which provides the access method that enables one to send email programmatically from SAS.

As an option, email addresses and other email options like “To” list, “CC” list, and attachment etc. can be defined in the statement.

The following is a typical email with report attached. This is an email we sent weekly to users to report adverse events and adjudication results for reconciliation. Note that if the output file is empty, the email content will be different and there will be no attachment in the email.

In the program, macro variables “&nob” and “&nob2” are the number of entries in two worksheets in the attachment Excel file. If both equal to zero, the Excel file will not be attached to the email. If “&nob” is zero, the email content will show as “Study1 AE QC Check Report has been updated. No mismatch is found across all AE forms for all patients”.

```
%MACRO _output;

%LET email_lst="jennifer@ABC.com" "marie@ABC.com" "tatyana@ABC.com";

FILENAME mymail email to=(&email_lst)
  from="Stephen, Smith <Stephen.Smith@ABC.com>"
  replyto="Frank, Fan <kang.fan@ABC.com>"
  subject=" Study1 AE QC Check Report, &dt"
  %if (&nobs gt 0) or (&nobs2 gt 0)
  %then attach=("&outpath\Study1 AEQC Check &dt..xls" );

DATA _null_;
FILE mymail;
PUT 'Hi all,';
PUT " ";
%IF &nobs gt 0 %THEN put "Study1 AE QC Check Report has been updated.
  Attached is a copy of the latest Study1 AE QC Check Report.";
%ELSE put "Study1 AE QC Check Report has been updated. No mismatch is found
  across all AE forms for all patients";
put " ";
put "Generated from Study1 AEQC system on &dtm";
```

```

RUN;

%MEND _output;

%_output;

```

For more information and detail about sending email from SAS, please refer to Fan 2011.

## CHECK ERROR/WARNING IN SAS LOG FILE

If we run a single program, we open the log file to check if there are any errors or warning messages in it, but if we run many programs in a batch file, it is troublesome to check all the logs one by one. Fortunately, we can leverage Windows' system functions and DOS commands to automate it. As the first batch file example shows, we can specify the location of log files. This batch file will proceed to check the log files to make sure that the first batch file is executing all the SAS programs properly.

Here is a simple code snippet to find errors/warnings in logs. The batch files are located in the same directory with the log files generated from SAS program execution. When it is executed, the "findstr" command will scan all the .log files to find the parameter specified, either "ERROR" in first line, or "WARNING" in second line. Then through ">", which is called "pipe", it redirects the output to the "find\_err" or "find\_warn" text files. "find\_err.txt" and "find\_warn.txt" give the user the error/warning summary found in all the logs scanned.

```

findstr ERROR: *.log > find_err.txt
findstr WARNING: *.log > find_warn.txt

```

## FIND THE LATEST FILE/FOLDER IN A DIRECTORY TO USE AS INPUT

In a lot of cases, we need to use the latest files in a directory as input, or we even save the files in different sub-folders to organize our data. Additionally, if the folder is created each time, we need to find the latest folder first. With the two macros below, one can automate these tasks easily.

To find the latest file, this macro can be used, which is derived from SAS Notes 42964:

```

%Macro Find_latest_file(fid, fname);
  /*Retrieve the result of "dir" command for files with modified date, in
  reversing order by date*/
  filename flst pipe "dir ""&fname."" /t:w /a:-d /o:-d /b";

  /*Execute the command and output to a dataset*/
  data dirlist;
    infile flst truncover;
    input fname $char1000.;
    put fname=;
  run;

  filename flst clear;

  /*Identify the file you need, if it is the 1st one, then &fid=1 */
  data latest_file (keep=fname);
    set dirlist(obs=&fid.);
  run;

  /*Make it a macro variable */
  %global NFile;
  data _null_;
    set latest_file;
    call symput("NFile",strip(put(fname,$char1000.)));
  run;
  %put &NFile;
%Mend;

```

Similarly, to find the latest folder, use this macro:

```
%Macro Find_latest_dir (did, dir);
/*Retrive the result of "dir" command for directories with modified date,
in reversing order by date*/
filename dlst pipe "dir ""&dir."" /t:w /a:d /o:-d /b";
/*Execute the command and output to a dataset*/
data dlist;
    infile dlst truncover;
    input name $char1000.;
    put name=;
run;
filename dlst clear;
/*Identify the directory you need, if the 1st one, then &did=1 */
data latestdir (keep=name);
    set dlist(obs=&did.);
run;
/*Assign the file name to the variable to return*/
%global NDir;
data _null_;
    set latestdir;
    call symput("NDir",strip(put(name,$char1000.)));
run;
%put &NDir;
%Mend;
```

## CONCLUSION

Automating SAS programs was a challenge for years and useful to clinical trial conduction. In the paper, we discussed the use of batch files with SAS programs, the application of scheduled task of Windows in SAS, and sending emails from SAS to further automating the process. Additionally, some miscellaneous automation tips are presented. The more powerful part is the combination of these tools in our daily work. With the emergence of these new technologies, we can go further in saving SAS programmers' time from menial tasks, allowing them focus more on data analysis and data science fields.

## REFERENCES

- MicroSoft website <https://technet.microsoft.com/en-us/library/bb490869.aspx>
- SAS 9.3 Language Reference: Concepts and SAS 9.3 Language Reference at <https://support.sas.com/documentation/cdl/en/lrcon/65287/PDF/default/lrcon.pdf>
- Fan, Frank. 2011. "Sending Emails in SAS® to Facilitate Clinical Trial". *Proceedings of the Western Users SAS Software 2011 Conference*. San Francisco, CA. Available at [http://www.wuss.org/proceedings11/Papers\\_Fan\\_F\\_74971.pdf](http://www.wuss.org/proceedings11/Papers_Fan_F_74971.pdf)
- Smoak, Carey. 2015. "Managing the Evolution of SAS Programming". *Proceedings of the Pharmaceutical SAS User Group 2015 Conference*. Orlando, FL. Available at <http://pharmasug.org/proceedings/2015/CP/PharmaSUG-2015-CP03.pdf>
- Michael B. Spring. "UNIX Scripts and Job Scheduling". Available at [http://www.sis.pitt.edu/spring/courses/Unix/Unix\\_6\\_Scripts\\_S.pdf](http://www.sis.pitt.edu/spring/courses/Unix/Unix_6_Scripts_S.pdf)
- SAS Notes 42964, "Sample 42964: Retrive the last modified file in a directory". Available at <http://support.sas.com/kb/42/964.html>

## CONTACT INFORMATION

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

Frank (Kangmin) Fan  
Abbott Vascular  
3200 Lakeside Drive  
Santa Clara, CA 95054  
Email: [frank.fan@av.abbott.com](mailto:frank.fan@av.abbott.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.