

Creating a DOS Batch File to Run SAS® Programs

David Franklin, IQVIA

ABSTRACT

We often have many SAS programs to run in a directory. While it is possible to run each individually, it is better if a DOS Batch file be created with the list of programs being run and the order in which they are run.

This paper looks at a SAS macro that will take the list of SAS programs in a directory, as specified by the user, and create a DOS Batch file to which can then be run to run all the SAS programs. Also presented will be a small SAS program that you can run at the end of the program to send you an email saying when the programs had finished, and whether there are any issues in the SAS LOG to review!

INTRODUCTION

There are 750 SAS programs in the directory. You could sit there all day and select Batch Submit from Windows Explorer, but that would be a waste of your time. Would it not be better to create a DOS Batch file and set that running? Also, at the end would you not like to know when the batch job finished and report any issues that need your attention?

CREATING THE DOS BATCH FILE

Often I have a number of SAS programs to run so I create a batch file in Windows and then run that batch file. But how do you create that batch file? As nearly always, there is more than one way to do it.

Lets first look at a SAS program which reads in a directory listing of SAS programs (those with the extension .SAS) and then creates a batch file with all the programs.

The first step of the process is to switch on the option NOXWAIT and set the directory we are interested in running the programs, as shown below:

```
%let dir=%str(directory_being_processed);
options NOXWAIT;
```

An example of such a directory_being_processed value could be

```
C:\TUAI\LISTINGS
```

Next step is to create a list of .SAS files and put that in the directory_being_processed directory with the file name of TMP.TXT -- these are in name order, using the option 'o:n' and exclude the directory name in front of the file using the option 'b'.

```
data _null_;
  length _txt $200;
  _txt='dir /b /o:n "||strip("&dir")||'\*.sas" > "||strip("&dir")||'\tmp.txt";
  put _txt=;
  call system(_txt);
run;
```

The following step reads in the names of the SAS programs.

```
data _lst0;
  length fn $200;
  infile "&dir.\tmp.txt";
  input fn;
  fn=upcase(FN);
run;
```

In the above step it is possible to have file name selection clauses after the last statement to exclude any programs that you do not want added to the batch file.

The last step creates the batch file, with the name of the file being specified in the FILE statement:

```
data _null_;
  file "&dir.\RUNALL.bat";
  length logfn txt $200;
  set _lst0 end=eof;
  if _n_=1 then do;
    put "set SASLOC=C:\Program Files\SAS\SAS 9.1";
    txt='set PRJLOC='||strip("&dir");
    put txt;
    txt=scan("&dir",1,"\");
    put txt;
    put 'CD %PRJLOC%';
    put 'DEL /Q *.LOG';
    put 'DEL /Q *.LST';
  end;
  txt="&SASLOC%\sas.exe" -rsasuser -noicon -nosplash '||
    '-nostatuswin -nologo -SYSIN "%PRJLOC%\'||strip(fn)||"';
  put txt;
  if eof then put 'EXIT';
run;
```

It is important to note that the line

```
put "set SASLOC=C:\Program Files\SAS\SAS 9.1";
```

sets the location of the SAS.EXE file which is different for site, operating system and SAS version -- the SET statement is a DOS command that allows a DOS variable to contain a value. A similar task is done in the next line

```
txt='set PRJLOC='||strip("&dir");
```

where the location of the directory being processed is being set.

The following lines will set the drive where your programs are and location for the programs to run:

```
txt=scan("&dir",1,"\");
put txt;
put 'CD %PRJLOC%';
```

Lastly, any LOG and LST files in the directory are deleted using the DOS DEL command:

```
put 'DEL /Q *.LOG';
put 'DEL /Q *.LST';
```

Note that as a rule I delete any LOG files before I run a SAS program just in case the (rare) situation arises where the SAS LOG is not replaced during the run of the program.

The lines

```
txt="&SASLOC%\sas.exe" -rsasuser -noicon -nosplash '||
  '-nostatuswin -nologo -SYSIN "%PRJLOC%\'||strip(fn)||"';
put txt;
```

create the DOS SAS batch file statement that calls each program.

Finally, the

```
EXIT
```

command closes the DOS window when the batch file is run.

An example of the file that is produced is shown below:

```

set SASLOC=C:\Program Files\SAS\SAS 9.1";
set PRJLOC=C:\TUAI\LISTINGS;
CD %PRJLOC%
DEL /Q *.LOG DEL /Q *.LST
"%SASLOC%\sas.exe" -rsasuser -noicon -nosplash -nostatuswin -nologo -SYSIN "%PRJLOC%\L1-DM.sas"
"%SASLOC%\sas.exe" -rsasuser -noicon -nosplash -nostatuswin -nologo -SYSIN "%PRJLOC%\L2-AE.sas"
EXIT

```

EMAILING IF THE PROGRAMS RAN OKAY OR HAD ISSUES?

To determine if a SAS program ran okay, it is important to review the SAS LOGs for the run. This is best done via a SAS log checker, an example of which is given below:

```

*Define location of LOG Files to process;
%let logdir=%str(C:\TUAI\PROGRAMS\ADS);

*Bring in ALL SAS LOG(s);
data _alllogs0;
  attrib _txt length=$256 informat=$char256. format=$char256.
         _fn length=$256
         _myinfile length=$256
         _ln length=8
         _myinline length=8;
  infile "&logdir.\A*.log" /*Here use t*.log for all table LOGS
                          but can be anything*/
         lrecl=256 filename=_myinfile line=_myinline length=len;
  input _txt $varying256. len;
  _fn=scan(strip(_myinfile),-1,'\');
  _ln=_n_;
proc sort data=_alllogs0;
  by _fn _ln;
run;

*Output findings;
title1 "QC of SAS LOG(s)";
data _null_;
  retain _k 0 anyissue 0; *Internal counter and if issue flag;
  file PRINT;
  set _alllogs0 end=eof;
  by _fn _ln;
  if first._fn then do;
    put @@_k1 '***** LOG FILE: ' _fn /;
    _k=0;
  end;
  if index(_txt,'ERROR:') or
     index(_txt,'WARNING:') then do;
    _k+1;
    put _txt;
    anyissue=1;
  end;
  if last._fn and _k=0 then put @1 '** NO ISSUES FOUND **';
  if eof then do;
    put / '/*EOF*/';
    call symput('anyissue',put(anyissue,1.));
  end;
run;

```

The code above will look for the words 'ERROR:' and 'WARNING:' in the directory C:\TUAI\PROGRAMS\ADS – if they are found then the lines that contain these words is output to the listing, else if no issues are found then the message "*** NO ISSUES FOUND ***" is displayed.

To add the piece that will email if any issues were found, and if so what they were, it is important to know if any issues were found – this is the ANYISSUE macro value above, 0 if none, 1 if issues. Now incorporate this into an email and we would use the following code:

```

filename outmsg email;
data _null_;
  file outmsg
    to="Huey@company.com"
    cc=("Dewey@company.com" "Louie@company.com")
    bcc="Della@company.com"
    subject="Run of ADS for &sysdate9.&&sysptime."
    attach = "C:\TUAI\PROGRAMS\ADS\SASQCLOG.LST";
  put 'Dear All';
  put;
  put 'Attached is SASQCLOG report for the TUAI Listings.';
  put;
  if "&anyissue"="1" then
    put "Issues were found in the SAS LOG and needs review.";
  else put "No issues were found in the SAS LOG.";
  put;
  put 'MySelf and I, ';
  put 'Lead Programmer';
run;

```

Finally, the code for the SAS LOG checker and email could be placed in a single SAS program, the DOS Batch file amended to call that program, and when run the programs run in sequence, a check of the SAS LOGs is carried out, and finally an email sent to people on the result.

CONCLUSION

Creating a DOS batch file solves an issue in making certain all programs are run in a directory and in the correct order. Also added is the ability to check the SAS LOGs for issues, and email the result to a group.

CONTACT INFORMATION

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

David Franklin

david.franklin1@iqvia.com

Any brand and product names are trademarks of their respective companies.