

Paper 105-27

Run All Your SAS® Programs in One Program: Automatically

Ling Y. Chen, Rho, Inc., Newton, MA
Steven A. Gilbert, Rho, Inc., Newton, MA

ABSTRACT

After SAS programs have been developed in an interactive computer environment, running them together in batch can save time and avoid errors. This paper describes how to use SAS to create a .BAT file that automatically lists and runs all SAS programs in a specified directory. A macro is presented that extends this procedure to an unlimited number of directories. The codes included in this paper were tested using SAS V6.12 on the Windows 98 operating system.

INTRODUCTION

For a clinical trial study, there are usually three stages of programming needed to create the necessary output for inclusion in a clinical trial study report. The first stage is called the "development" stage, during which we write the individual SAS programs and test them for bugs. This stage, which typically uses a "first cut" of the study data, may be quite lengthy if there are significant changes to the database prior to locking. The second stage occurs after final database lock when the programs are run again and put through one last round of quality control. The quality control procedure used at this point is quite extensive and includes a manual review of all program logs. The programs will not be allowed to have any error messages and all warning messages will be checked and documented. Finally, when the programs have successfully completed the in-house quality control process we are ready for the third and final "production" stage. During the production stage it is usually desirable to produce a final copy of the output that has the same date stamp on all documents for the clinical study report.

Depending on the size of the study, there are sometimes dozens or even hundreds of programs that need to be run at the production stage. Physically opening up, sitting through the run, and then closing each program can be tedious and time consuming. It is also quite likely that either a necessary program is not run or some sequence-dependent programs are not run in the proper order. Fortunately, these problems can be avoided by running the SAS programs in batch.

RUNNING SAS PROGRAMS IN BATCH

To run a SAS program in batch, you need to have a .BAT file. To create the .BAT file, first open a new text file using Notepad and then save it with the .BAT extension. If you want to run the program C:\chen\prog1.sas, you need to create a .BAT file that reads:

```
C:\SAS\SAS.exe -sysin C:\chen\prog1.sas
```

Note that this code assumes your SAS is installed in the C:\SAS directory. You may need to adjust the path to your particular setup.

To run multiple programs sequentially, this code needs to be modified as below:

```
Start/w C:\SAS\SAS.exe -sysin C:\chen\prog1.sas
Start/w C:\SAS\SAS.exe -sysin C:\chen\prog2.sas
Start/w C:\SAS\SAS.exe -sysin C:\chen\prog3.sas
```

This .BAT file, when double clicked, will run C:\chen\prog1.sas first, then C:\chen\prog2.sas, and finally C:\chen\prog3.sas.

GENERATING A LIST OF ALL SAS PROGRAMS UNDER A FOLDER NAME

The structure of the .BAT file dictates that there be one line of code containing the name of the program for each program you want to run. If you want to run a hundred programs in a batch, you need to enter a hundred lines of code. This process is clearly tedious if you have to do it manually. Furthermore, after the file is created, you need to make sure that you have included all the programs you want to run, spelled the names of the programs correctly and entered the correct path. To avoid all of these problems, we use a DOS command to generate the list of programs. The following DOS command generates a .TXT file containing the names, file extensions, and other information of all of the files in that folder. For example,

```
dir c:\ >proglst.txt
```

will generate a text file called proglst in the C:\ directory. This list can be generated in SAS by adding an X in front of the DOS command:

```
x "dir c:\ >proglst.txt"
```

Note, however, that this command creates a file in the C:\ directory. If you want to generate a file in the same directory as the programs, then use the following code:

```
x "dir c:\chen >c:\chen\proglst.txt"
```

The generated .TXT file looks like the following:

```
Volume in drive C has no label
Volume Serial Number is 07CF-0B10
Directory of C:\chen

.                <DIR>          05-24-01  2:35p  .
..               <DIR>          05-24-01  2:35p  ..
PROG1  SAS        0  05-24-01  2:35p  prog1.sas
PROG2  SAS        0  05-24-01  2:35p  prog2.sas
PROG3  SAS        0  05-24-01  2:35p  prog3.sas
PROG1  LST        0  05-24-01  2:36p  prog1.lst
PROG2  LST        0  05-24-01  2:36p  prog2.lst
PROG3  LST        0  05-24-01  2:36p  prog3.lst
PROG1  LOG        0  05-24-01  2:36p  prog1.log
PROG2  LOG        0  05-24-01  2:36p  prog2.log
PROG3  LOG        0  05-24-01  2:36p  prog3.log
      9 file(s)                0 bytes
      2 dir(s)                10,628.16 MB free
```

AUTOMATIC GENERATION OF THE .BAT FILE USING SAS

Now that we have a text file containing the names of all the files in a particular folder, we can use it in a data step to generate the .BAT file needed for batch submission of the SAS programs. Here are the codes for the data step:

```
Filename proglst 'c:\proglst.txt';
```

```
data progs;
  infile proglst;
  input @1 var1 $ @10 var2 $;
run;
```

```
data progs;
  set progs;
  char="start/w c:\sas\sas.exe -sysin
c:\chen";
  var3=char||compress(var1)||'.sas';
  where var2='SAS';
run;
```

Note that VAR1 captures the names of the files and VAR2 contains the extension of the files so that files with a .SAS extension can be selected. VAR3 is the concatenated character variable that makes up the content of the .BAT file. The end result is a data set containing a line of code containing the name of the program for each program in that directory.

CREATING ONE .BAT FILE FOR MULTIPLE PROGRAMS IN MULTIPLE FOLDERS

The preceding codes can easily be modified into a macro to account for multiple programs in multiple folders.

```
%macro genlist (folder=, dsn=);

x "dir &folder > &folder\proglst.txt"

Filename proglst "&folder\proglst.txt";

data progs;
  infile proglst;
  input @1 var1 $ @10 var2 $;
run;

data &dsn;
  set progs;
  char="start/w c:\sas\sas.exe -sysin
&folder";
  var3=char||compress(var1)||'.sas';
  where var2='SAS';
run;

%mend genlist;

%genlist (folder=C:\chen, dsn=chen);
%genlist (folder=C:\gilbert, dsn=gilbert);
```

Each time the macro is called, a data set is generated containing the list of programs for a particular folder. This can be repeated for as many folders as desired. These data sets are then concatenated and written to a single .BAT file.

```
Filename runprog "C:\RUNPROG.BAT";

data all;
  set chen gilbert;
run;

data _null_;
  file runprog;
  set all;
  put @1 var3;
run;
```

The resulting .BAT file is:

```
Start/w C:\SAS\SAS.exe -sysin C:\chen\prog1.sas
Start/w C:\SAS\SAS.exe -sysin C:\chen\prog2.sas
.
.
.
Start/w C:\SAS\SAS.exe -sysin C:\gilbert\prog1.sas
Start/w C:\SAS\SAS.exe -sysin C:\gilbert\prog2.sas
.
.
.
```

This .BAT file should be reviewed to ensure that any sequence-dependent programs are in the proper order.

CONCLUSION

Once the .BAT file has been generated, it is just a matter of double clicking it in order to run your programs one after another without any human intervention. A task that would otherwise require a programmer, a spare computer, and hours of waiting time is now just two clicks away. Before you leave work, double click to start it, and by the time you arrive at work the next day, you will have your programs nicely run with the same date stamp. You can also schedule the event to occur at a time that is most convenient and efficient, (i.e., 2 o'clock in the morning). It is important to remember that there will not be any human supervision of the .log files. Therefore, it is a good idea to use Windows, by right clicking on the program folder and using the search dialog box, to search through the .log files for error and warning messages, even though all programs have been through final QC.

REFERENCES

The following reference represents selected material from the SAS Institute web site and is a useful supplement to this document:

- <http://ftp.sas.com/techsup/download/technote/ts648/ts648.pdf>

SAS is a registered trademark of the SAS Institute Inc. in the USA and other countries.

ACKNOWLEDGMENTS

We would like to thank Katherine Monti, Ph.D. for her comments on this paper.

CONTACT INFORMATION

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

Ling Yun Chen
Rho, Inc.
199 Wells Ave, Suite 302
Newton, MA 02459
617-965-8000 x26
Fax: 617-965-8014
lchen@rhoworld.com