

BAT Files: Run all Your Programs with One Click in PC SAS

William Conover, Advanced Clinical, Bannockburn, IL

ABSTRACT

This is a simple, fast and easy way to automatically create a BAT file that runs all of the SAS® programs in one directory.

INTRODUCTION

In PC SAS 9.1, the ability to run all SAS programs in one directory with a click of the mouse is a powerful tool for efficiency. This ability is also very easy to accomplish with the reusable 'drag and drop' type program presented in this paper.

The basic concept is to first use a macro copied from the SAS support website, %DRIVE(), that lists all the program names within a directory. The second step is to use DATA _NULL_ and the FILE statement to output a BAT file containing syntax to run all the programs. The final step is to actually use the file to run all the SAS programs in the directory. With that, a few custom modifications may be desired and are discussed.

For the sake of simplicity, all the code discussed here is placed in a single program named 'Create_BAT.sas'. This program can be placed in any directory, run in batch mode by right clicking on the file and selecting the appropriate option, and will create the file '_Run_All_SAS_Programs.bat' that runs all SAS programs in the current directory, excluding 'Create_BAT.sas'.

STEP 1: LIST ALL THE SAS PROGRAMS IN A DIRECTORY

Creating a list of SAS programs in a directory can be accomplished a multitude of ways using different utilities, macros, or simply hand entering each program name into a spreadsheet, text file or macro variable. For this application, a macro copied from the SAS support website, %DRIVE(), is utilized. This macro lists files with a user defined extension, in this case '.sas', and prints the list to the SAS log file. With a slight modification, this list is redirected to a temporary text file named 'Create_BAT.txt'. The modified code is presented below:

```
filename prgnm 'Create_BAT.txt';          *<-- text file w/ Program names;
libname here ' ';                        *<-- location of local Programs;

data _null_;
  call symput('herepth',pathname('here'));*<-- pathname for %drive macro;
run;
*--MACRO DRIVE-----*;
%macro drive(dir,ext);
  %let filrf=mydir;
  %let rc=%sysfunc(filename(filrf,&dir));
  %let did=%sysfunc(dopen(&filrf));
  %let memcnt=%sysfunc(dnum(&did));
  proc printto log=prgnm new;           *<-- direct the log output to the text file;
  run;
  %do i = 1 %to &memcnt;
    %let name=%qscan(%qsysfunc(dread(&did,&i)),-1,.);
    %if %quppercase(%qsysfunc(dread(&did,&i))) ne %quppercase(&ext) %then %do;
      %if (%superq(ext) ne and %quppercase(&name) = %quppercase(&ext)) or
        (%superq(ext) = and %superq(name) ne) %then
        %put %qsysfunc(dread(&did,&i));
    %end;
  %end;
proc printto log='Create_BAT.log';*<-- log output to log file;
```

```

run;
%let rc=%sysfunc(dclose(&did));
%mend drive;
*--end: MACRO DRIVE-----*;

```

The macro call looks like this:

```
%drive(&herepth,sas); *<-- create list of programs as text file;
```

...where the macro variable **&herepth** is defined by the LIBNAME 'here';

%DRIVE() POST PROCESSING

Some post processing of the text file 'Create_BAT.txt' will be necessary to create a clean list of program names since it will still have a few lines of log information in it. Sample code to remove the excess information is below:

```

Data table;
  infile prgnm delimiter='!';*<-- read in the text file 'Create_BAT.txt';
  length ttlvar $135.;
  input ttlvar $;
run;

data table;
  set table;
  if index(ttlvar, '.sas')>0;*<-- keep text lines with .sas extension;
  prog=scan(ttlvar,1, '.');
  if index(prog, '_BAT')>0 then delete;*<-- omit the program name 'Create_BAT.sas';
  drop ttlvar;
run;

```

Now the dataset WORK.TABLE has a list of program names stored in the variable TTLVAR.

STEP 2: CREATING THE BAT FILE

To create the BAT file, a new dataset WORK.SASPROGS is created containing the variable LOCSAS, which is simply a text string of the directory path combined with the program name.

```

data sasprogs;
  set table;
  sort1=_n_;
  progsas=trim(left(prog))||'.sas';*<-- add the .sas extension;
  locsas=pathname('here')||'\'||trim(left(progsas));*<-- add the directory path;
  keep locsas;
run;

```

Next, DATA_NULL_ is used to output the text string containing path and program name, along with the necessary batch processing code.

```

%let SAS_dir="c:\Program Files\SAS\SAS 9.1\sas"; *<-- dir where PC SAS is, keep
the '\sas' portion on the end;

data _null_;
  set sasprogs;
  FILE '_Run_All_SAS_Programs.bat' ;
  PUT @1 &SAS_dir " -batch -uprint -uprintmenuswitch -nosplash -icon -SYSIN "
    "'locsas'";
  put ;
run;

```

The above DATA_NULL_ creates a file named '_Run_All_SAS_Programs.bat' in the current directory.

Note that the macro variable &SAS_DIR will need to be modified according to the current PC SAS installation. To locate this directory, simply look for the file 'SAS.EXE' in a directory similar to 'c:\Program Files\SAS\SAS 9.1', then copy and paste that path into the macro variable definition for SAS_DIR, retaining the current quotation marks. Be sure to add '\sas' to the directory path, this runs the file 'sas.exe' during batch processing.

STEP 3: RUNNING THE BAT FILE

Double click on the file '_Run_All_SAS_Programs.bat', or right click and select 'Open'. A MS-DOS window will appear and display text related to the running of each SAS program. When all SAS programs have run, the MS-DOS window will close. To prematurely terminate the running of the BAT file, first close the MS-DOS window by clicking the 'X' in the upper right hand corner, and then maximize the window with the SAS icon. There should be options to terminate the SAS job in that window.

TIPS

- Running programs in batch mode while an interactive PS SAS session is open will typically cause WARNING messages in the log. A good practice is to quit interactive PC SAS before running the BAT file.
- The BAT file can be edited with any text editor. Removing or adding filenames manually gives this method great flexibility for running programs.
- If programs should be run in a specific order, name the program files so they are in the correct order alphabetically in the directory, or simply edit the BAT file manually with a text editor.
- The batch file processing options '-nosplash' and '-icon' suppress the display of the SAS logo screen and minimizes the individual batch window for each program respectively. If either of these two items is desired during batch processing, simply remove the desired statement from the file '_Run_All_SAS_Programs.bat' with a text editor, or in the DATA_NULL_ section in 'Create_BAT.sas'.
- Customize the BAT file by editing '_Run_All_SAS_Programs.bat' to run only certain types of programs (i.e. Tables, Listings, Figures, Datasets, etc.). Although, be sure to rename the BAT file accordingly so it won't be overwritten the next time 'Create_BAT.sas' is run (ex: '_Run_All_Tables.bat').

FULL PROGRAM: CREATE_BAT.SAS

```
options mprint notes linesize=146 ps=48 pageno=1 nodate orientation=landscape
nocenter;          *--- standard options;

*****
** Company:  Advanced Clinical Research Services
** Author:   wconover
** Date :    10-2-08
** File Type: Utility
** Production Use: Yes
** Validation: Extensive use across multiple study protocols with QC review.
**
** Purpose:  creates the _Run_All_SAS_Programs.bat file
**
** Notes: user needs to define directory where PC SAS is installed (indicated as:
MODIFY HERE)
**
** Modifications:
**  author/date:
**  description:
**
*****;

*---MODIFY HERE-----*;

%let SAS_dir="c:\Program Files\SAS\SAS 9.1\sas"; *--- keep quotes, directory where
PC SAS is installed, keep the '\sas' portion on the end;

*---end: MODIFY HERE-----*;

*****
** Company:  Advanced Clinical Research Services
** Author:   SAS website (http://support.sas.com/kb/35/344.html)
```

```

** Date :      7-29-08
** Purpose: This macro enables you to print all files or certain files with a
particular extension to the SAS log.
**          This macro accepts 2 parameters. The first parameter is the directory
that contains the files.
**          The second parameter is the extension of the files you want to search
for.
**
** Notes: Leave 2nd paramater blank if you want a list of all the files.
**
** sample MACRO call:
**          %drive(c:\,sas)
**
** Modifications:
** author/date: wconover/7-29-08
** description: modified output destination
**
*****
filename prgnm 'Create_BAT.txt';
          *<-- file w/ Program names;
libname here ' ';
                                     *<-- location of local Programs;

%macro drive(dir,ext);
  %let filrf=mydir;
  %let rc=%sysfunc(filename(filrf,&dir));
  %let did=%sysfunc(dopen(&filrf));
  %let memcnt=%sysfunc(dnum(&did));

  proc printto log=prgnm new;
  run;

  %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
          %put %qsysfunc(dread(&did,&i));
          %end;
%end;

  proc printto log='Create_BAT.log';
  run;

  %let rc=%sysfunc(dclose(&did));
%mend drive;
*--end: MACRO DRIVE-----*;

*-----*
  List all Program Names
*-----*;
data _null_;
  call symput('herepth',pathname('here'));
run;
                                     *<-- pathname for %drive macro;

%drive(&herepth,sas);
  *<-- create list of programs as text file;

*----*;

Data table; infile prgnm delimiter='!';
  length ttlvar $135.;
  input ttlvar $;
run;

data table; set table;

```

```

        if index(ttlvar, '.sas') > 0;
        prog = scan(ttlvar, 1, '.');
        if index(prog, '_BAT') > 0 then delete;
        drop ttlvar;
run;

*-----*
create
  _Run_All_SAS_Programs.bat
*-----*;
data sasprogs; set table;
  sort1 = _n_;
  progsas = trim(left(prog)) || '.sas';
  locsas = pathname('here') || '\' || trim(left(progsas));
  keep locsas;
run;

*----output BAT file-----*;
data _null_;
  set sasprogs;
  FILE '_Run_All_SAS_Programs.bat' ;
  PUT @1 &SAS_dir " -batch -uprint -uprintmenuswitch -nosplash -icon -SYSIN "
  "'locsas'" ;
  put ;
run ;

*----clean-up-----*;
data _null_;
  fname = "tempfile";
  rc = filename(fname, "Create_BAT.txt");
  if rc = 0 and fexist(fname) then
    rc = fdelete(fname);
  rc = filename(fname);
run;
*----end: clean-up-----*;

```

REFERENCES

SAS website Sample 35344: How to read all XML files in a directory with the XML92 LIBNAME engine - <http://support.sas.com/kb/35/344.html>

ACKNOWLEDGMENTS

Special thanks are due to the Advanced Clinical Biostatistics team for the review and feedback of this paper and SAS programming techniques.

CONTACT INFORMATION

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

Name:	William Conover
Enterprise:	Advanced Clinical
Address:	1200 Lakeside Drive
City, State ZIP:	Bannockburn, IL, 60015
E-mail:	wconover@acrxs.com
Web:	http://www.advancedclinical.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.