Let the system do the work! Automate your SAS code execution on UNIX and Windows platforms

Niraj J. Pandya, Element Technologies Inc., NJ
Vinodh Paida, Impressive Systems Inc., TX

ABSTRACT

Often, same SAS® macros are run to produce specific SAS outputs on a regular basis and many times on a strict scheduled time in a production environment. Automating this process can save lots of time and resources. The whole process can be streamlined. This paper explains how to automate the SAS jobs which will execute the required SAS macros on a given specific time point, will also check the log for errors and warnings. If there are any errors, it will send an e-mail with log file; else if the job is successful then it will send the SAS output to the e-mail to appropriate personnel. This article explains about the automation on Microsoft Windows® platform using the task scheduler as well as CRONTAB facility on UNIX® platform.

INTRODUCTION

Whether you have to run daily, weekly or monthly reports or you have to generate incremental reports on a specific day of the week and on specific time, or you have to extract the data right at the midnight, you have to be there in front of the machine to run your jobs. Such time sensitive tasks are always prone to some possible issues like you forgot to run the job or you are running late to work or internet connection has some issues and connectivity is lost etc. For whatever reason you are not there to take care of such regular tasks, wouldn't it be nice if system can remember such tasks and execute it for you on time and also can take care of task related post-execution activities like sending the message if report is failed or send a report as an attachment to appropriate person within the team if job is successful?

By creating a batch (.bat) file and associating it with Task Scheduler, Windows operating system can do the job for us. Similar is possible on UNIX platform also using CRONTAB utility.

WINDOWS TASK SCHEDULER

Windows task scheduler is one of the system tools on Windows operating system where you can schedule multiple tasks and windows operating system will perform those tasks on a specified time and schedule.

For illustration purpose, let's say that you want to run a specific SAS code called test.sas on a specific day and time of a week. Let's say that you want this SAS code to run on every Sunday at 12:00 p.m. To setup this task, you need to log in to Windows server and follow the steps as specified below.

To setup the task scheduler in Windows environment, you need to setup a batch file first.

BATCH FILE

A batch file is a text file containing a series of commands intended to be executed by the command interpreter. Similar to job control language systems on mainframe and UNIX systems, batch files in windows systems are there to ease the work required for certain regular tasks by allowing the user to set up a batch script to automate many commands. When a batch file is run, the shell program (usually cmd.exe) reads the file and executes its commands, normally line-by-line. Batch files are useful for running a sequence of executables automatically and are often used to automate repetitive or tedious processes. In DOS and Windows environment, batch files usually have filename extension as .bat.

To create a batch file, open notepad and add the entries as instructed below and save the file with .bat extension. Let's call it sample.bat for illustration purpose in this paper. You can write the similar batch file from SAS code also.

For example, the batch file entry can be some thing like below:

start/w C:\"Program Files"\SAS\v9\sas.exe –sysin *SAS code location***\test.sas -NOSPLASH -PRINT ***SAS output location***\test.lst -LOG ***SAS log location***\test.log**

The above mentioned entry will be interpreted as run the test.sas file from specified location using SAS without opening a SAS window and direct and test.lst and test.log files in to specified location respectively.

You can have multiple entries in this batch file if you want multiple SAS codes to run and in case that you want them to run in a specific order, batch file entries have to be in the same order. You can also edit this batch file in future to add more entries or to remove some entries.

STEPS TO SCHEDULE A TASK

Once the batch file setup is done, follow the steps specified below to associate the batch file to the task scheduler to execute it on a specific time.

A.   Go to control panel and double click 'Scheduled Tasks' icon which will then popup a window as shown in below image.
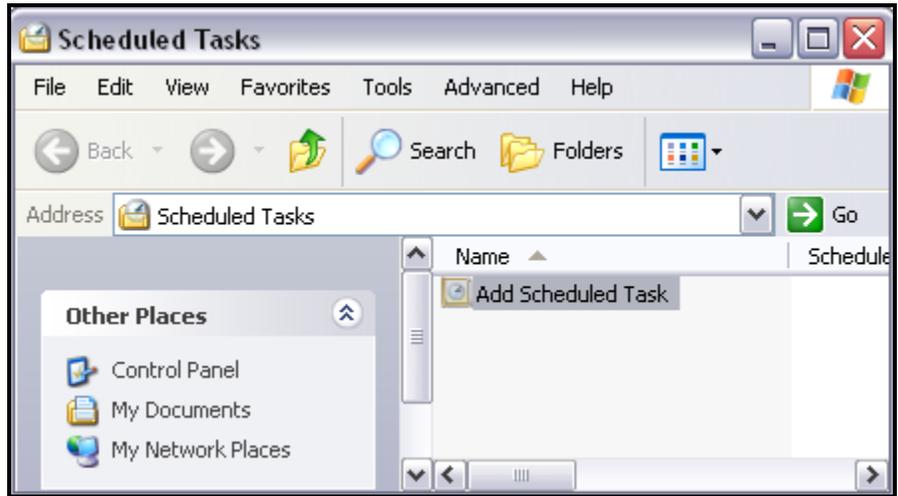

Image 1

B. Double click the 'Add Scheduled Task' icon and it will take you to the next screen. There click on the 'Next >' button which will open a window as shown in below image.
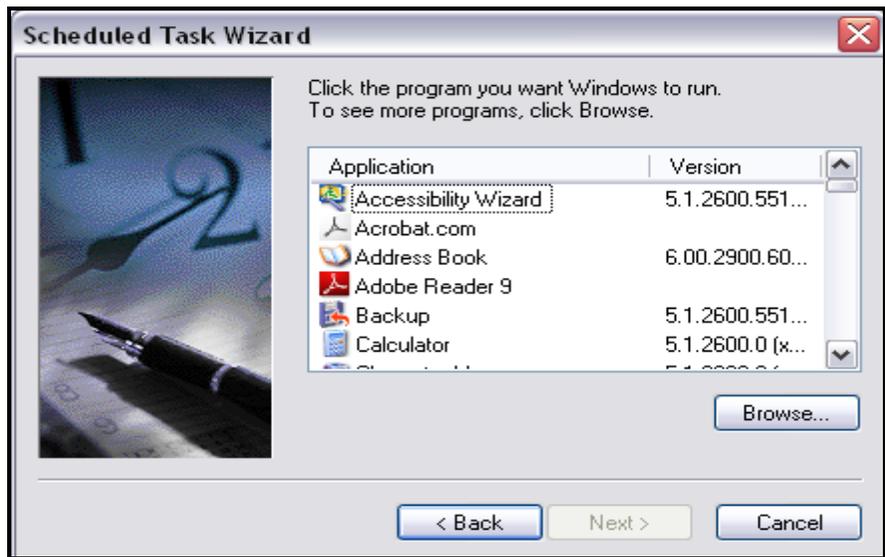

Image 2

C. Click on the 'Browse...' button and select the batch file from where it is located and click the 'Open' button.

D. Give some name you wish for the scheduled task and click on the 'Weekly' radio button as shown in below image if you want the batch file to be executed every week which in turn will run test.sas every week. Then click 'Next>' button. In this paper we will call this task as 'Weekly run of test.sas' as we want to run it on every Sunday at 12:00 p.m.

Image 3

The options on next window depend on the selection you make on this current window.

If you select 'Daily', next screen will give you options of start time, perform the task everyday or only on weekdays or on the interval of some days and the start date from which this task has to be performed.

If you select 'Weekly', next screen will give you options of start time and the day of the week. Here you can also select more than one day in a week.

If you select 'Monthly', next screen will give you options of start time, day of the month and different months in a year. Here also you can select more than one month based on your schedule requirement.

If you select 'One time only', next screen will ask you for start time and star date only when you want to perform the task.

For remaining two selections, there won't be any more options and next screen will directly ask for username and password.

E.   Now select the time and day of the week when you want the task to be run.

Image 4

F.   It will then ask you to enter your user name and the password. Enter your domain\user name and the password and click 'Next > ' button. You will see the window as shown in below image which will confirm the details of the scheduled task.

Image 5

G. Finally click 'Finish' button. That's it; you are all set and ready to go. If you also select the check box for advanced properties on this window, it will open new window when you click 'Finish' button.

This new window gives you more control for your scheduled task. It shows you different active and inactive tasks that you have scheduled along with the actual schedule and also gives you options on how to handle the task when it is running longer than some specific time or system is running on batteries or scheduled task occurs when system is idle and many more other options.

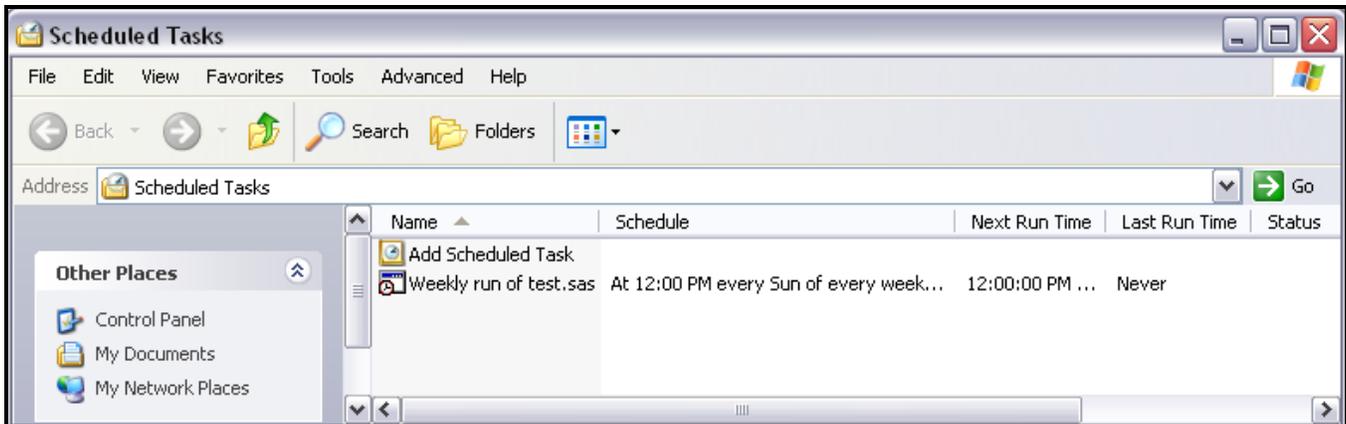Now, if you perform step A again, you will see your scheduled task in the list there as shown in image below.

Image 6

There is one limitation when you want to use windows task scheduler to execute your SAS codes automatically on a given time. If you are running the task scheduler on a stand alone PC, the system has to be ON on a given time for task scheduler to perform the tasks.
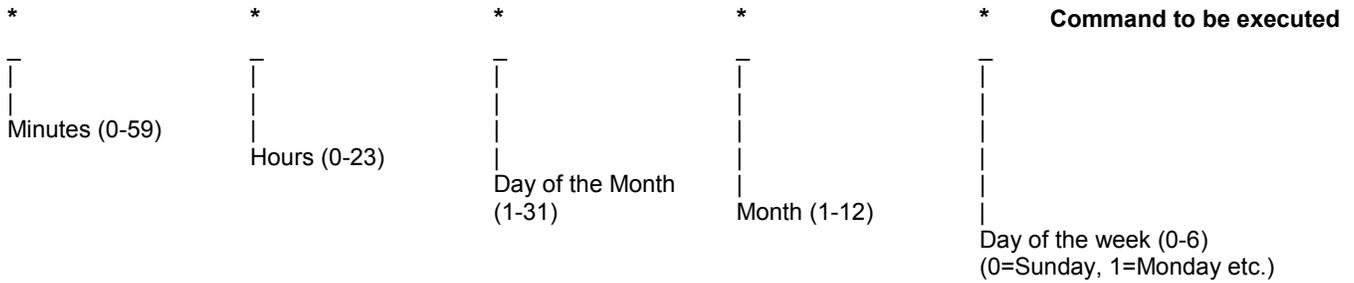
UNIX CRONTAB UTILITY

CRON is a utility in UNIX that allows tasks to be run automatically in background on specific times and intervals by cron daemon. CRONTAB is the file that holds a list of commands that are to be run at specific time in the background. It is a file that contains the schedule of entries or commands to be run at specified times and intervals. CRON always invokes the commands from user's home directory.

To use this utility, you just have to add the entries to your crontab file. Some basic UNIX commands to see and edit your crontab file are as below.

crontab -e	Edit your crontab file or create one if it doesn't already exist.
crontab -l	Display your crontab file entries.
crontab -r	Remove your crontab file.
crontab -v	Display the last time you edited your crontab file. (This option is available on few systems only)

To do this, open a UNIX terminal and enter crontab –e command. An editor will open displaying your current crontab file. Edit the crontab file using the below format as described. Save the changes and exit. It's that simple!

The structure of crontab file is explained below. A crontab file has five fields separated by space for specifying day, date and time followed by the command to be run at that time point.

```
*            *            *            *            *      Command to be executed
_            _            _            _            _
|            |            |            |            |
|            |            |            |            |
Minutes (0-59)  |            |            |            |
            Hours (0-23) |            |            |
                     Day of the Month  |            |
                     (1-31)       Month (1-12)  |
                                            Day of the week (0-6)
                                            (0=Sunday, 1=Monday etc.)
```

In above five fields, a value of '*' means all allowed values for that field. If you have to specify more than one value for any of the field, you can do that by separating two values by comma (,). If you want to specify a range, you can do that using hyphen (-).

Now let's say we want to run our test.sas code on every Sunday at 12:00 p.m. For this task, crontab file entry will be:

00 12 * * 0   /usr/local/bin/sas /*SAS code location***/test.sas**

Once you edit your crontab file with the entry as mentioned above, you just have to save the file and exit and rest will be taken care by cron daemon. You can always edit this entry or add more entries in future using basic crontab command and all your scheduled run job of SAS code will be in one place.

Now that we have looked in to the structure of crontab and how it works, next we will look in to how a SAS code can handle crontab file and how SAS code can handle the email of output based on the result of cron job in next section of this paper.

SAS MACRO TO SCHEDULE A TASK

A SAS code can add and edit entries in crontab by including some basic x commands in the code. Following macro code explains a simple way to do so.

```
%macro my_schedule;

 X crontab -l > /home/someuser/myschdl;

 Data crontab;
     Infile '/home/someuser/myschdl' pad;
     Length text $200;
     Input text $1-200;
 Run;

 Proc sql;
      Create table cnt as select *
      from &syslast;
 Quit;

 %LET NOOFOBS = &sqlobs;
```

```
   Data new;
         Text = "00 12 * * 0 /usr/local/bin/sas /SAS code location/test.sas";
         Output;
   Run;

   Data myschdl;

         %if &NOOFOBS = 0 %then %do;
           Set new;
         %end;

         %else %do;
           Set crontab new;
         %end;

         If text = '' then delete;
   Run;

   Proc sort data = myschdl nodupkey;
         By text;
   Run;

   Data _null_;
         Set myschdl;
         File "/home/someuser/myschdl";
         Put text;
   Run;

%mend;


X crontab /home/someuser/myschdl;

X rm /home/someuser/myschdl;


%my_schedule;
```

Above macro first reads the crontab file and copies all the entries in to a temporary file named 'myschdl' using x command. Then it reads this temporary file in to SAS dataset named 'crontab' and all entries will be stored as values of variable named TEXT. If there are no entries in your current crontab file then this dataset will be empty. Then this macro creates another dataset named 'new' and all new entries that you want to add in your crontab file are added in this dataset, again variable name is TEXT.

If dataset 'crontab' is not empty then a new dataset named 'myschdl' is created by appending datasets 'crontab' and 'new'. But if 'crontab' dataset is empty then 'myschdl' dataset will be created from only 'new' dataset. Then macro deletes any duplicate entries from the dataset 'myschdl' and writes this dataset again in to previously created temporary file named 'myschdl'.

Finally, in the end, again using x command, this macro writes the crontab file using updated temporary file 'myschdl' and also deletes this temporary file.

This simple macro either creates a crontab file with your specified entry or it updates the already existing crontab file with new entry.

Above mentioned macro will update the crontab file so that test.sas will be run on every Sunday at 12:00 p.m.

Now, the purpose here to use cron utility is you want to run some SAS code and create some report when you are not logged in to the system or you are away from the system. But you are still interested in knowing the result of test.sas, whether it ran successfully or it encountered an error. You can add following piece of code in to test.sas and based on the outcome of this SAS program, you can email either the log file with error or list file with the report to yourself or any one else that you want to notify of the outcome.

In the beginning of the test.sas, add below mentioned code to get the program name in a macro variable and direct your log and list files to a specific location using this macro variable.

```
%LET pgmname = %sysfunc(getoption(sysin));
%LET pgm = %scan(&pgmname,1,'.');

%put &pgmname;
%put &pgm;

 Proc Printto print = "&pgm..lst" new;
 Proc Printto log = "&pgm..log";
```

Then add following code at the end of test.sas to email either log or list file based on the outcome of this SAS program.

```
%If &syserr = 0 %then %do;

  X cat &pgm..lst | mailx -s "SUCCESS: &pgmname has been run successfully" user@xyz.com;

%end;
%else %do;

  X cat &pgm..log | mailx -s "FAILURE: &pgmname has produced an error" user@xyz.com;

%end;
```

Above mentioned code will check if there are any errors occurred while running test.sas. If there is no error, it will send the test.lst file in the email to specified email address with subject line mentioning about successful run of test.sas. If there are any errors occurred, then it will send test.log file in the email to specified email address with subject line mentioning about failure while running test.sas.

CONCLUSION

With the use of already available system utilities, SAS codes which are required to be run repeatedly can be executed automatically on required time and interval. User is not required to be physically present in front of the system. Such scheduled SAS run jobs will run in the background without interrupting user's current work and no user input is required at that particular time point. SAS code and algorithm explained in this article is for example purpose and shows a simpler way of achieving the result. Similar result can be achieved using many alternate algorithms and user can add many more functionalities using example from this article as a stepping stone.

CONTACT INFROMATION

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

> Name: NIRAJ J PANDYA
> Phone: 201-936-5826
> E-mail: npandya@elementtechnologies.com

> Name: VINODH PAIDA
> Phone: 860-333-4178
> E-mail: Vinodh.saspro@yahoo.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.