Paper BB-02

While You Were Sleeping, SAS® Was Hard At Work

Andrea Wainwright-Zimmerman, Capital One Financial, Inc., Richmond, VA

ABSTRACT

Automating and scheduling SAS code to run over night has many advantages, but there are also many pitfalls to be aware of. This paper will discuss how to make sure you save the log, get e-mail messages at critical points, as well as other recommendations and considerations on how to make it all work.

INTRODUCTION

There are many advantages to scheduling SAS code to run after typical business hours. It can avoid tying up your resources while you have other tasks to do. If the code pulls data from a database, performance may be better after hours when there are fewer other queries running.

Nevertheless, not just any SAS code can be scheduled to run unattended. The code needs to be completely bomb proof with no syntax errors. There needs to be thorough error handling for unavoidable errors. In addition, the log should be saved in case something still goes wrong. E-mail notification of progress and results is essential. The last consideration is the actual scheduling of the code which will depend largely on the platform

BOMB PROOFING THE CODE

The code should be completely free of syntax errors and should have been run many times before planning to schedule it to run after hours. It should have been run in many different scenarios so that any possible errors that can’t be avoided are known and those that can be avoided have been avoided.

ERROR HANDLING

No matter how well written the code, there may still be errors. Three main questions to consider are:

1. Where in the code can errors happen?
2. What method can be used to detect the errors?
3. How should the errors be handled?

Some examples of unavoidable errors would be bad data as an input or database issues.

BAD DATA

If it is possible to receive bad data that will cause errors in the code, there must be detection code to check for problems. An e-mail with a clear subject line and helpful information in the body is recommended. The code that will produce errors due to bad data should be skipped. GOTO and LINK are helpful. If there is nothing else useful to do ENDSAS is also an option.

DATABASE ISSUES

Another unavoidable error would be if the database is down, or if the password has expired. One way to check for this programmatically is calling the following macro immediately after the libname statement that connects to the database:

```
%macro check_libname;
  %if &sysdbrc ne 0 %then %do;
    %send_email(subj=LIBNAME FAIL
               ,text=%superq(sysdbmsg))
    endsas;
  %end;
%mend;
```

The system macro SYSDBRC has the return code for the database connection. If it is 0, the connection was successful. Any other value indicates an issue. The system macro SYSDBMSG contains the text of the error message you would see in the log. It will indicate what the problem was. The details of the send_email macro are included later in the paper.

STORING THE LOG

No matter how well written the code, and how well planned the error handling, there still may be the occasional error. Full disk drives or other hardware issues are always possible and you can't realistically know when to expect them, how to detect them, or what to do to avoid it. The log should be stored so that it can be reviewed if a problem has occurred. This is simple to achieve with PROC PRINTTO:

```
proc printto log="C:\My Documents\project\log_&sysdate..log" new;
run;
```

The system macro SYSDATE will have the current date. By adding this to the file name the log won't be overwritten every day.

CONFIRMING SUCCESSES

At certain points in the process, the code should send out e-mails, not only if there are failures, but about successes too. If there is a macro that will be iterating several times, it is helpful to get an e-mail once for every iteration with a subject line mentioning that it is iteration X of Y. This allows you to watch for anomalies such as receiving a message 2 of 2 without receiving one about 1 of 2. This would be an indication you should check the log.

SEND_EMAIL MACRO

The following code is a simple macro for sending an e-mail. It assumes that the person normally running it has established a macro variable called my_e_mail with his or her own e-mail address. By default, the macro will send the message to, from and will carbon copy this e-mail. In this way, a call to this macro can override the TO line, while still sending a copy to the person running it.

```
%macro send_email ( to=  &my_e_mail
                  , from=&my_e_mail
                  , subj=
                  , cc=  &my_e_mail
                  , text=
                  );

filename sendmail email;

data _null_;
    file sendmail;

   put '!EM_TO!' "&to";
   put '!EM_FROM!' "&from";
   put '!EM_CC!' "&cc";
   put '!EM_SUBJECT!' "&subj";
   body="&text";
   put body;
run;

%mend;
```

OUTLOOK® ISSUES

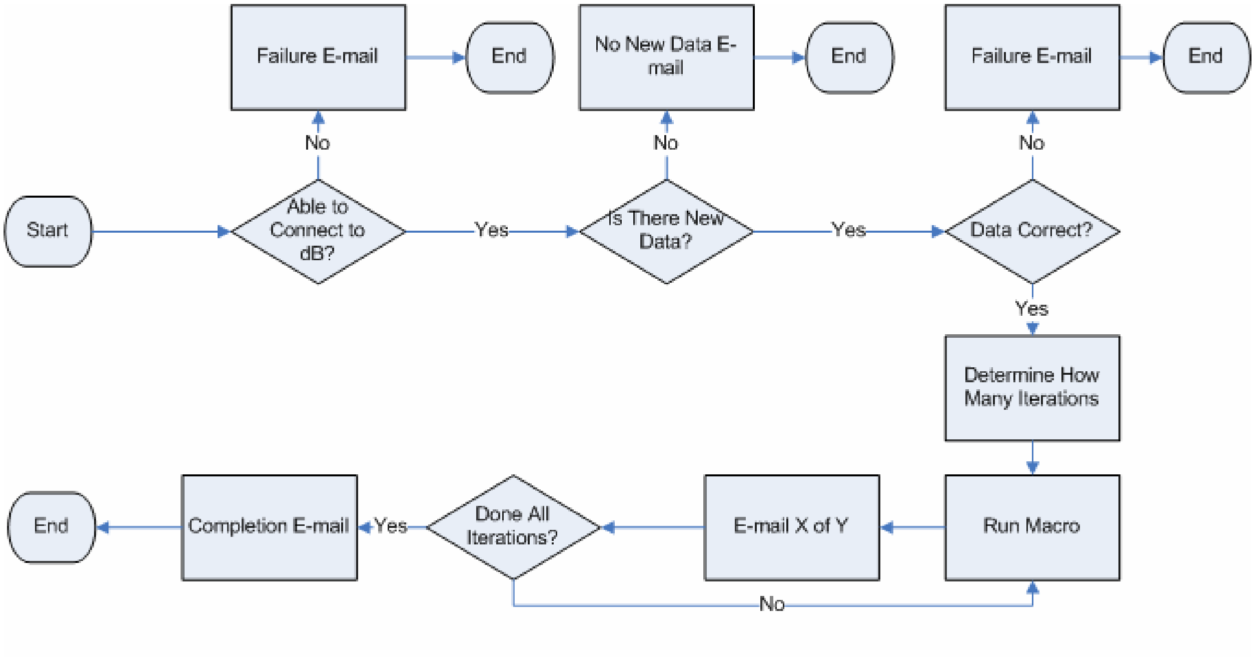
If Microsoft Outlook is the default mail program, then it will want to confirm if you really want the e-mail to be sent. A pop-up will appear and everything will wait for you to click yes. There are a few options for dealing with this.

1. It is an optional behavior of Outlook that can be turned off. However, typically permission to turn it off is limited. One needs either to have that permission, or have influence with those who do.
2. There are free programs that can be downloaded such as "Click Yes". The sole purpose of the program is to click the yes button that Outlook pops-up. However, many times permission to install such programs is tightly controlled.
3. The last option is to edit the SAS CFG file to use SMTP instead of Outlook. For more details see the official SAS site http://support.sas.com/kb/19/767.html.

DETERMINING SCENARIOS

There are probably many possible scenarios for the code to be scheduled. It may not always produce something; there may not always be anything for it to do. A simple flow chart can help determine what possible actions might be needed.

SAMPLE FLOW CHART



Don’t leave yourself wondering what happened!

SCHEDULING THE CODE TO RUN

Does the code need to run on a recurring basis, or does it just need to run after hours? If it just needs to run after hours, the SLEEP function can be used to delay it either for a set amount of time, or till a specific time. If it needs to run on a recurring schedule, the scheduling tool for the platform

SLEEP

To set it to run at a specific time, the number of seconds between now and the desired start time must be calculated. This code should be included early in the code. SAS will pause on the DATA _NULL_ till the calculated number of seconds has passed. Then it will proceed with the rest of the code.

```
data _null_;
   now=datetime();
   start=’25JUN2010:19:00:00'dt;
   wait_sec=start-now;
   zzz=sleep(wait_sec);
run;
```

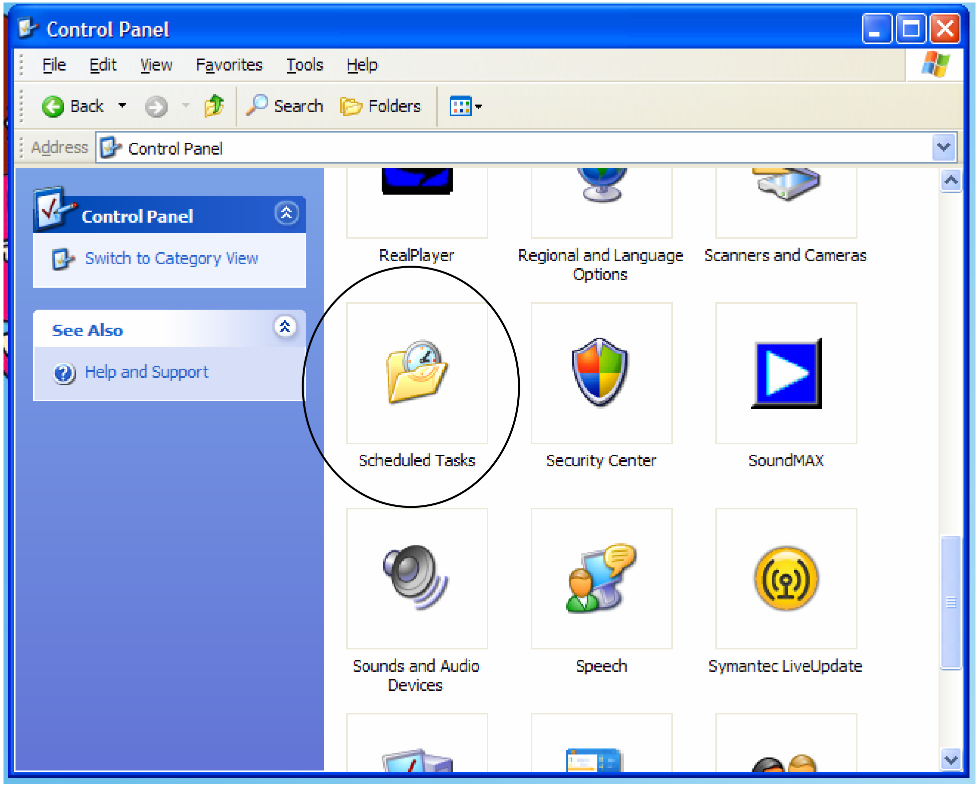
To just delay the code starting for a set amount of time, include the following code early. SAS will pause till the specified amount of time has passed before continuing with the rest of the code.

```
data _null_;
   hour=5;
   min=30;
   wait_sec=(60*60*hour)+(60*min);
   zzz=sleep(wait_sec);
run;
```

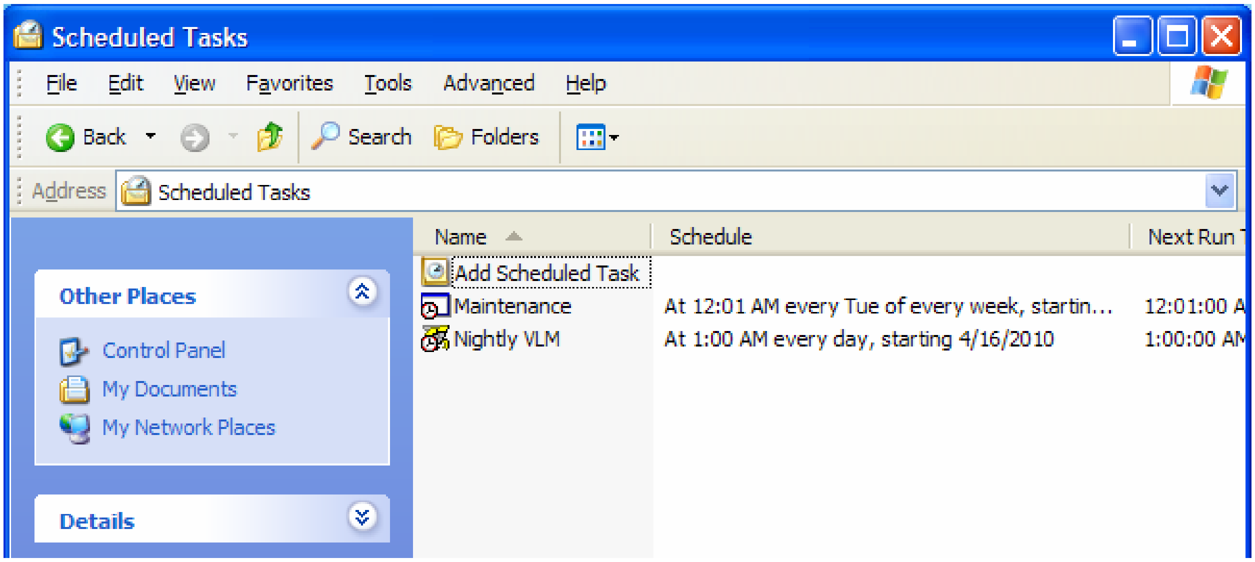
WINDOWS® SCHEDULED TASKS

If the code will need to run on a recurring basis then you will want to use whatever scheduling tools are available to your platform. For PC SAS users, Window Schedule Tasks can accomplish this goal.

Open the Control Panel and select the Scheduled Tasks icon.

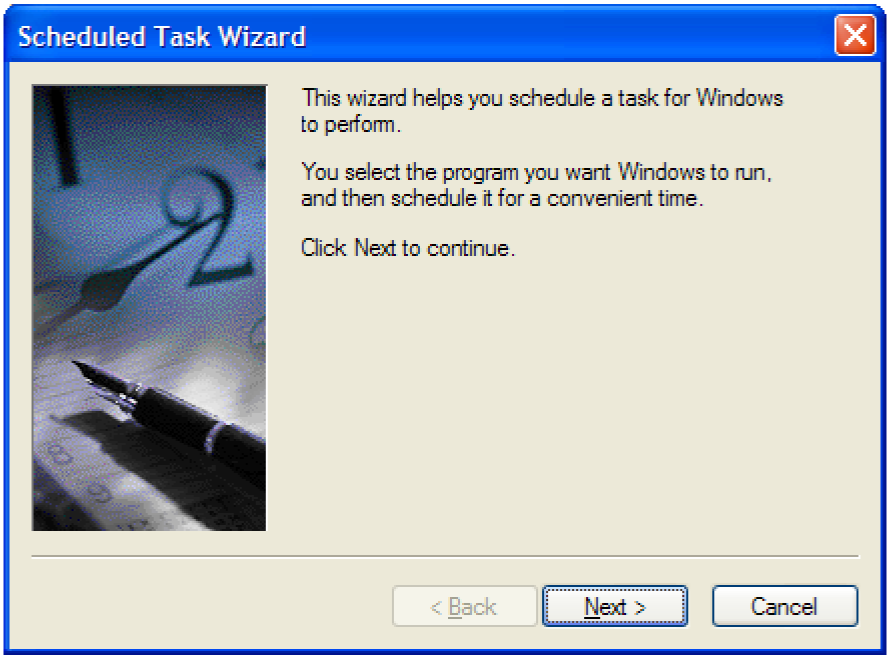


Click Add Scheduled Task to start the wizard.



Name	Schedule	Next Run
Add Scheduled Task		
Maintenance	At 12:01 AM every Tue of every week, startin...	12:01:00 A
Nightly VLM	At 1:00 AM every day, starting 4/16/2010	1:00:00 AM

Click Next.

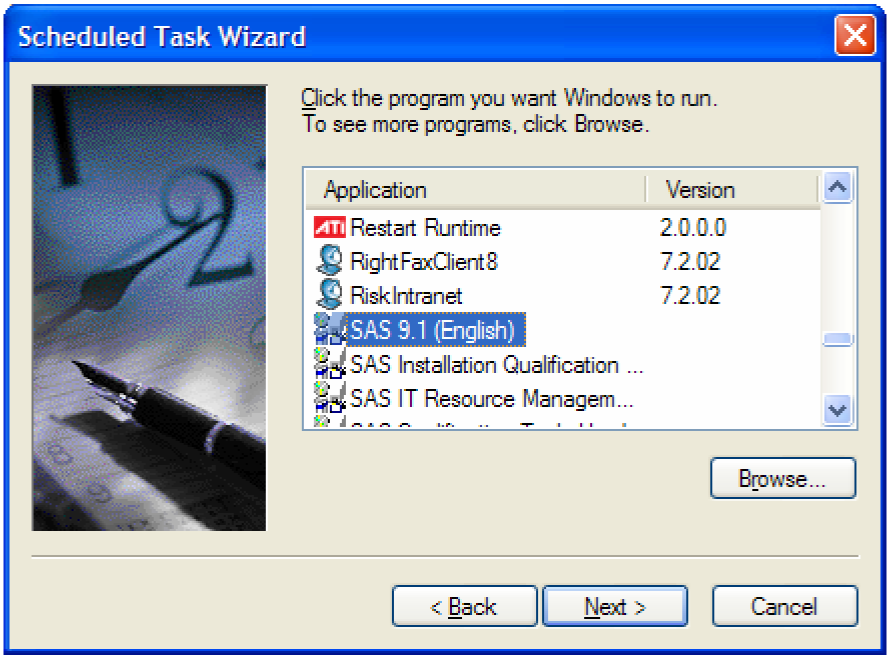


This wizard helps you schedule a task for Windows to perform.

You select the program you want Windows to run, and then schedule it for a convenient time.

Click Next to continue.

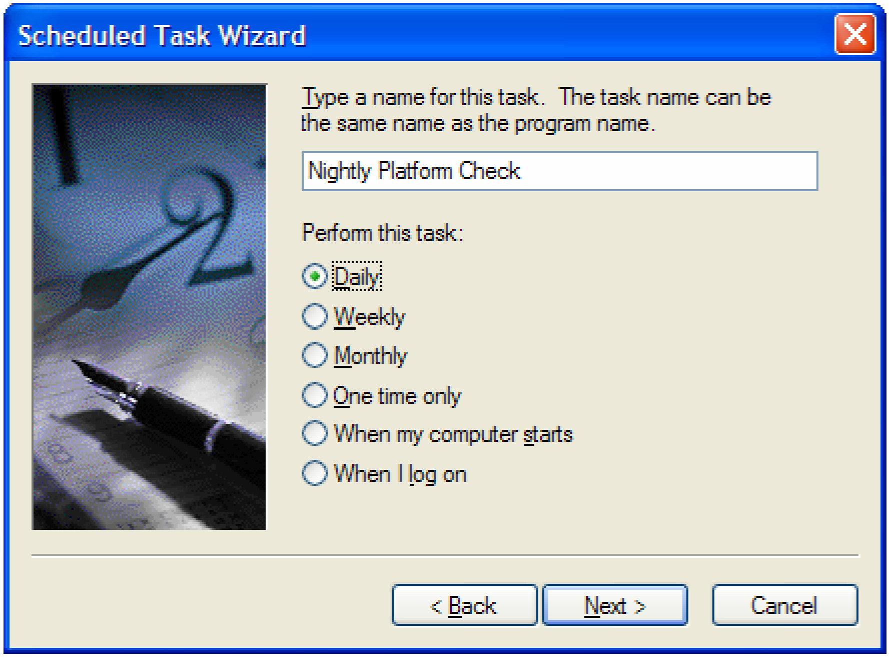
It may take some time for the wizard to compile a list of programs to select from. If you do no see your version of SAS, use the browse button to find the SAS.exe file.



Click the program you want Windows to run.
To see more programs, click Browse.

Application	Version
Restart Runtime	2.0.0.0
RightFaxClient8	7.2.02
RiskIntranet	7.2.02
SAS 9.1 (English)	
SAS Installation Qualification ...	
SAS IT Resource Managem...	

Name the task something meaningful and then select the frequency you want it to run.



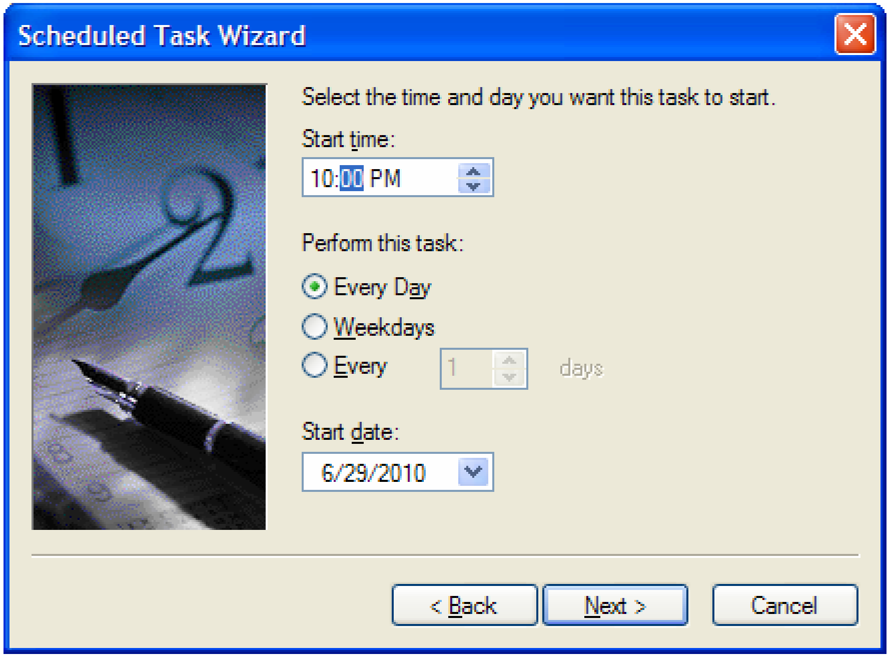
Type a name for this task. The task name can be the same name as the program name.

Nightly Platform Check

Perform this task:

● Daily
○ Weekly
○ Monthly
○ One time only
○ When my computer starts
○ When I log on

Set the time and other settings which will differ based on the frequency you choose.



Select the time and day you want this task to start.

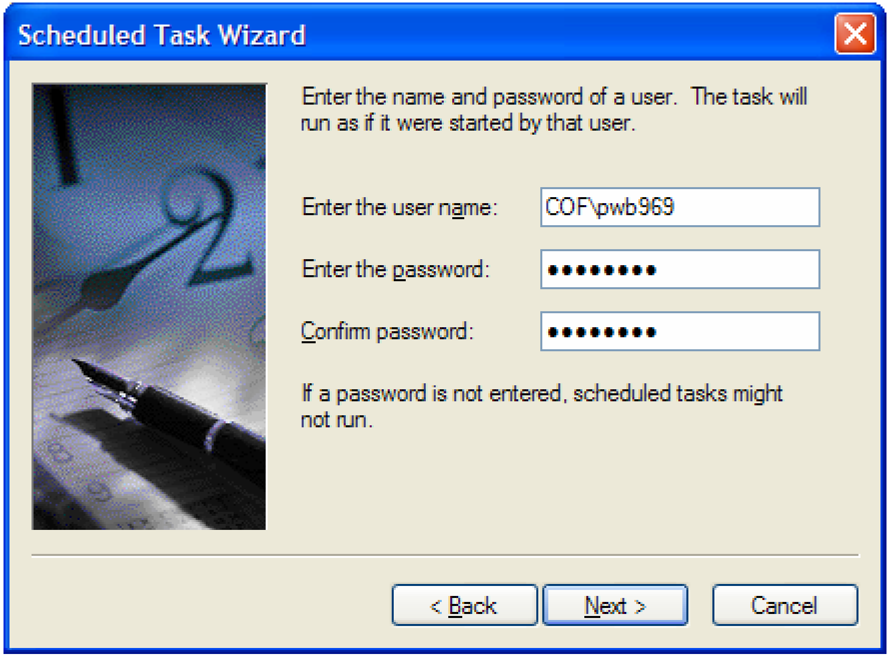
Start time:
10:00 PM

Perform this task:

● Every Day
○ Weekdays
○ Every 1 days

Start date:
6/29/2010

You will need to put in your user id and password so the computer can wake up to run the program.



Enter the name and password of a user. The task will run as if it were started by that user.

Enter the user name: COF\pwb969

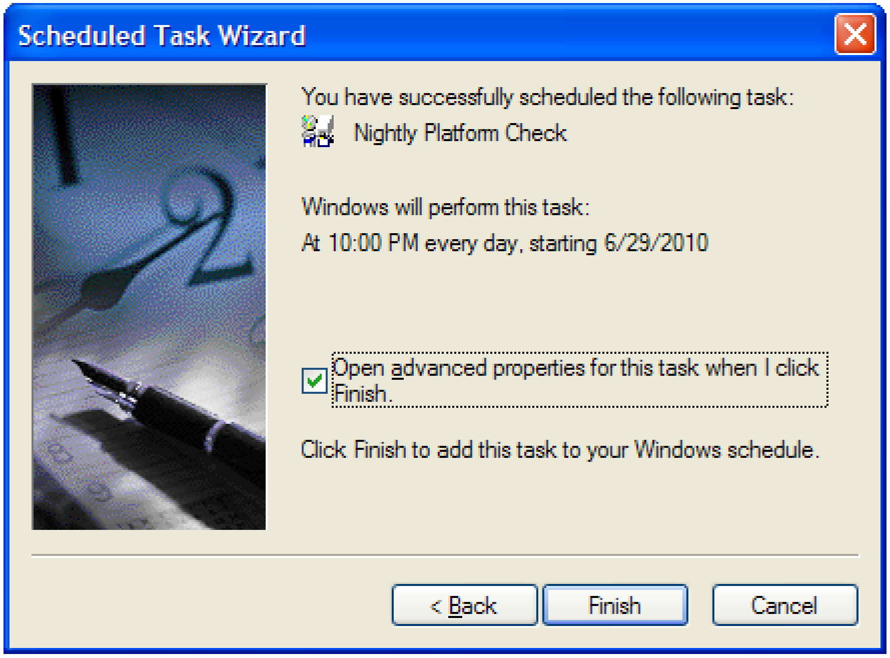
Enter the password: ●●●●●●●●

Confirm password: ●●●●●●●●

If a password is not entered, scheduled tasks might not run.

This will need to be updated anytime you change your computer password.

Check the details in the summary screen. Go Back and change anything if you need to. Make sure the check box to open the advanced properties is marked.



You have successfully scheduled the following task:

Nightly Platform Check

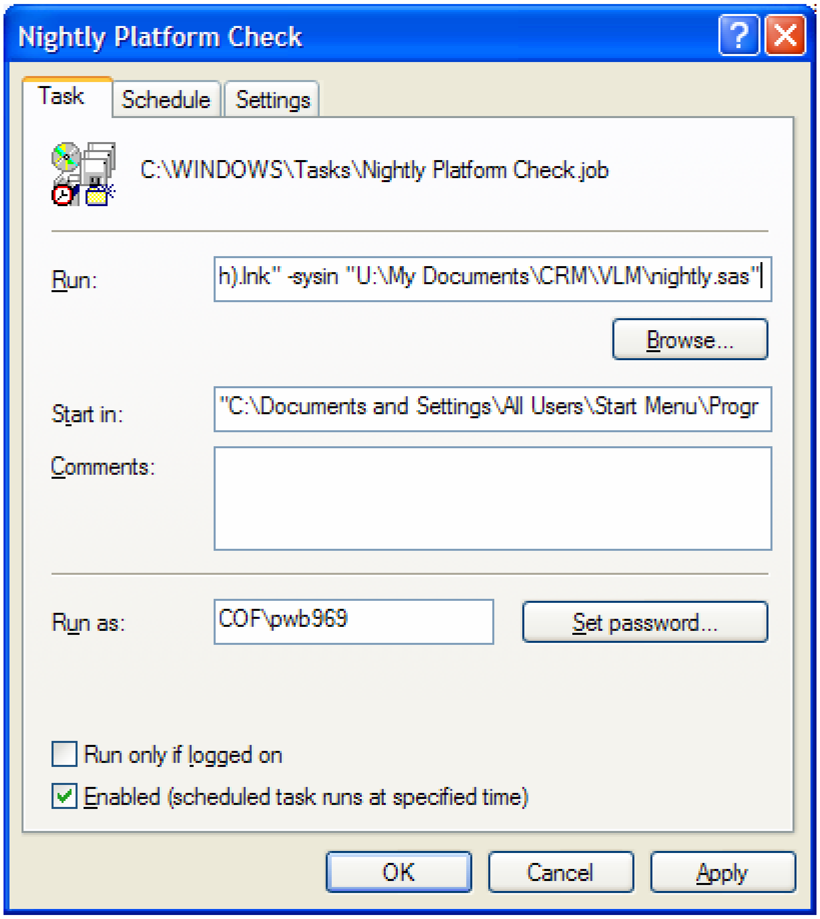
Windows will perform this task:

At 10:00 PM every day, starting 6/29/2010

☑ Open advanced properties for this task when I click Finish.

Click Finish to add this task to your Windows schedule.

Once the properties window comes up, make sure you are on the Task tab. Click in the Run: textbox, then move the cursor to the far right. Without removing anything already there, add a space, then “-sysin” followed by another space. Then, in quotes, enter the path to the specific sas program you want to run. Make sure the Enabled box is checked, otherwise it won’t run.



C:\WINDOWS\Tasks\Nightly Platform Check.job

Run: h).lnk" -sysin "U:\My Documents\CRM\VLM\nightly.sas"

Start in: "C:\Documents and Settings\All Users\Start Menu\Progr

Comments:

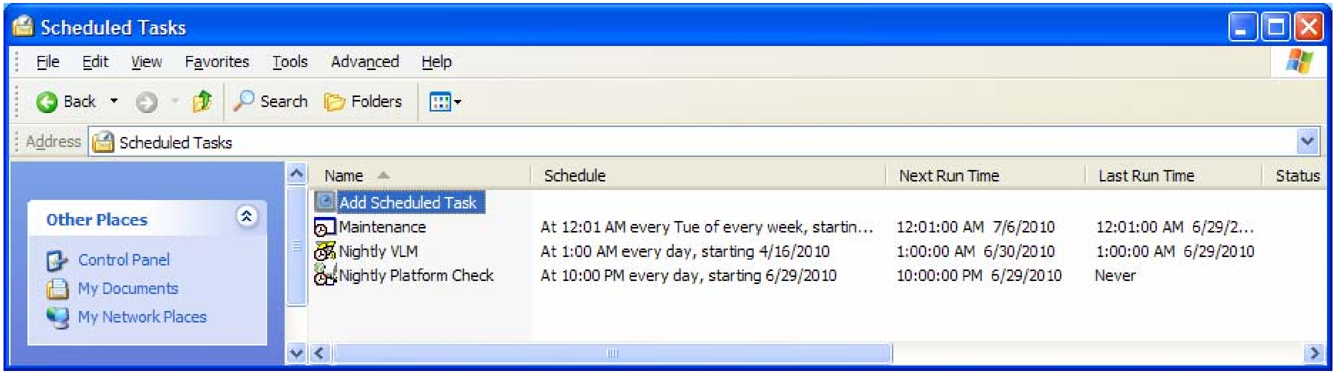
Run as: COF\pwb969

☐ Run only if logged on

☑ Enabled (scheduled task runs at specified time)

You can bring up this advanced properties window by double clicking the task in the Schedule Tasks folder at anytime, and the “Set password…” button above can be used to update the password as needed.

You will see the new task listed among your schedule tasks. You can see what the schedule is, when it plans to run it again, and when it was last run. In the case of the new task, “Never” is displayed.



Name	Schedule	Next Run Time	Last Run Time	Status
Add Scheduled Task				
Maintenance	At 12:01 AM every Tue of every week, startin...	12:01:00 AM 7/6/2010	12:01:00 AM 6/29/2...	
Nightly VLM	At 1:00 AM every day, starting 4/16/2010	1:00:00 AM 6/30/2010	1:00:00 AM 6/29/2010	
Nightly Platform Check	At 10:00 PM every day, starting 6/29/2010	10:00:00 PM 6/29/2010	Never	

CONCLUSION

There are many advantages to scheduling code to run after hours, either on an ad hoc basis for faster run times while not tying up resources during work hours, or to run on a regular basis without you having to worry about remembering to run it. There are some pitfalls and considerations before setting code to run, but coming in to work in the morning and having your reports ready, and an inbox with a few messages letting you know everything ran perfect is a great feeling.

ACKNOWLEDGMENTS

Thanks to all SAS-L participants for creating a wonderful, open place to ask questions and learn.

Thanks to the VASUG officers and members, past and present, for their help and support.

CONTACT INFORMATION

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

Andrea Wainwright-Zimmerman
Capital One Financial, Inc.
15000 Capital One Drive
Richmond, VA 23238
Work Phone: 804-284-7681
E-mail: andrea.zimmerman@capitalone.com