

## Better Ways to Speak to Your System Using SAS: Automate Routine Tasks by using X, SYSTASK & FILENAME

Ranganath Bandi, CliniRX Research (USA), Chicago, USA  
Harini Kunduru, Bristol Myers Squibb Company, Pennington, USA

### ABSTRACT

The main objective of this paper is to discuss how interactions with your operating system can be done using SAS® by triggering the operating system's commands to fetch required information to make your programming easier and more robust in nature. This paper mainly deals with usage of the X, SYSTASK and SAS FILENAME statements in combination with the PIPE device type to interact with the host operating system and how these statements can be used to attain the same results. Programmers will be guided through some of the system commands in both the UNIX® and Windows® operating systems environment which will help them to get a better understanding of how to make optimal use of the commands to automate certain routine tasks without having to re-do them manually each time and re-documenting.

This paper gives examples related to creating directories; copying existing files to new directories, fetching attributes of external files, and opening excel spreadsheets, CSV files, etc. in the host operating system. The examples provided are executed under SAS 9.1 under the Windows Vista and under UNIX operating system.

### INTRODUCTION

Many SAS programmers will be hesitant when it comes to routine work or when it comes to the documentation of the project after programming is completed for all the deliverables. Though this paper doesn't explain all the programming for how to automate these processes or writing a macro for doing the repeated work periodically, programmers will be given a basic idea on how to go further in using the operating system's commands as per their requirements.

### WHY TO USE AND WHERE TO USE

Generally, when we need to archive existing output or data that is present in a folder and create a new set of outputs and datasets we use the following process. We manually go to that particular folder, physically copy the files from that directory and place them in the archive directory that we manually created for archiving the old set of output/data. This can be done by using the CMD commands in Windows and Shell commands in UNIX, but even this need to be done manually either by directly running the shell scripts or by executing each of the commands. In this new scenario, we can write all the commands in SAS and incorporate them into a SAS program, so that the whole job can be done at one go.

This paper has examples for X, SYSTASK and PIPE under both the Windows and UNIX operating systems. The examples are limited to 1) Creating a new archive directory, 2) Archiving/Copying the existing set of outputs to the new archive directory, 3) Changing the file attributes of the copied files in the new archive directory, 4) Acquiring file attributes such as the date time stamp of the files, 5) Reading the file attributes to a SAS dataset, 6) Opening/Creating new Excel® or CSV files 7) Utility program to do the routine job and 7) Finally a conclusion and comparison between the X, FILENAME PIPE device type and SYSTASK.

Keywords: X, SYSTASK, FILENAME PIPE etc.

### HOST: WINDOWS OPERATING SYSTEM

| X STATEMENT  | PIPE DEVICE TYPE   |
|--|--|
| <pre>***USING NOXWAIT NOXSYNC*****; OPTIONS NOXWAIT NOXSYNC;</pre> <p>The option NOXWAIT specifies the command processor to return automatically to the SAS session after the specified command is executed. NOXSYNC is used for</p> | <pre>***1) To create a new directory for copying the existing set of outputs;  filename make pipe "MKDIR C:\pharmsug\progs\test3"; data _null_; infile make;</pre> |

executing the operating system commands asynchronously with your SAS session.

\*\*\*1) To create a new directory for copying the existing set of outputs;

```
X "MKDIR C:\pharmsug\progs\test2";
```

\*\*\*2) Copying output to the new directory;

```
X "COPY C:\pharmsug\progs\*.rtf
C:\pharmsug\progs\test2";
```

\*\*\*If you want to delete the existing files in the directory;

```
X "DEL /q C:\pharmsug\progs\*.rtf";
```

\*\*\*3) To change the attributes of the copied files in the new archived folder;

```
X "ATTRIB +r C:\pharmsug\progs\test2\*.*";
```

\*\*\*If you want to rename an existing file in the new folder;

```
X "RENAME
C:\pharmsug\progs\test2\l1_prog2.rtf
l2_prog2.rtf";
```

\*\*\*4) Now fetch the file attributes;

```
X "DIR C:\pharmsug\progs >
C:\pharmsug\progs\attributes.txt";
```

\*\*\*If you want to get even the owner of the output files;

```
X "DIR /q C:\pharmsug\progs >
C:\pharmsug\progs\attributes.txt";
```

For executing the multiple commands in one X command you can separate each one by '|'

```
X "ATTRIB +r C:\pharmsug\progs\test2\*.*
| del /q C:\pharmsug\progs\*.rtf " ;
```

```
run;
```

\*\*\*2) Copying output to the new directory;

```
filename copying pipe "COPY
C:\pharmsug\progs\*.rtf
C:\pharmsug\progs\test3";
data _null_;
    infile copying;
run;
```

\*\*\*If you want to delete the existing files in the directory;

```
filename delete pipe "DEL /q
C:\pharmsug\progs\test3";
data _null_;
    infile delete ;
run;
```

\*\*\*3) To change the attributes of the copied files in the new archived folder;

```
filename attrib pipe "ATTRIB +r
C:\pharmsug\progs\test3\*.*";
data _null_;
    infile attrib;
run;
```

\*\*\* If you want to rename an existing file in the new folder;

```
filename rname pipe "RENAME
C:\pharmsug\progs\test3\l1_prog2.rtf
l2_prog2.rtf";
data _null_;
    infile rname;
run;
```

\*\*\*4) Now fetch the file attributes;

```
filename attfile pipe "DIR C:\pharmsug\progs
> C:\pharmsug\progs\test3\attributes.txt";
data _null_;
    infile attfile;
run;
```

\*\*\*If you want to get even the owner of the output files;

```
filename attfile pipe "Dir /q
C:\pharmsug\progs >
C:\pharmsug\progs\test3\attributes.txt";
```

\*\*\*Reading file attributes like date time stamp for each file;

```
filename attfile pipe "Dir
C:\pharmsug\progs";
```

5) To read the file attributes into a SAS dataset by reading the text file attributes.txt created by the X statement, or the filename with PIPE device statement, or by fileref 'attfile' created by the filename statement.

```
%macro readatb(type=);
    %if %upcase(&type)=XCOMMAND or %upcase(&type)=PIPEEXT %then %do;
        Filename attfile "C:\testst\progs\attributes.txt";
    %end;
data demo;
    infile attfile trunccover pad lrecl=32367 length=len end=eof;
    input line $varying200. len;
    if index(upcase(line), ".RTF")>=1 ;
    moddate=scan(compbl(line),1,' ');
    time=scan(compbl(line),2,' ');
```

```
pgname=scan(compbl(line),5,' ');
run;
%mend readtab;
```

## 6) To trigger Excel by using the X command

```
X CALL "C:\Program Files\Microsoft Office\Office12\excel.exe";

filename xcel dde "EXCEL|SYSTEM";
data _null_;
  file xcel;
  put '[open("book1")]';
  put '[save.as("C:\testst\progs\summary\test.xls")]';
  x=sleep(1);
run;
```

The path reference to Excel .EXE will change as per the path of the .EXE file on your particular PC. You can export the file attributes which were read into the dataset demo to the Excel file created by using DDE Triplet. This is not explained in this paper as it is not in the scope of it.

After exporting the file attribute information to the Excel file, you can save it and close the application.

```
data _null_;
  file xcel;
  put '[save()]';
  put '[quit()]';
run;
```

For other directories, Example if you want to copy files from one directory named 'test listings' to testlistings then It needs to be specified in the following way:

```
X COPY "C:\test listings\*.rtf" "C:\testlistings";
```

Though it is not recommended to have directory names having spaces, even if they have they can be handled in this way.

## SYSTASK (WINDOWS)

SYSTASK by default executes the operating system commands asynchronously from within your SAS session or application, which means that these tasks are executed independently of all other tasks that are currently running.

\*\*\*1) Making a directory:

```
SYSTASK COMMAND "MKDIR C:\pharmsug\progs\test2" wait;
```

\*\*\*2) Copying output to the new directory;

```
SYSTASK COMMAND "copy C:\pharmsug\progs\*.rtf C:\pharmsug\progs\test2" WAIT
MNAME=mytask TASKNAME=SASJOB STATUS=TASK;
```

If we want each task to execute in sequence you have to use the option 'WAIT'. The MNAME option is used for specifying the macro variable with the name assigned to the submitted job and if the TASKNAME option is used copies the name you specified in TASKNAME to the macro variable specified in MNAME.

Systask gives a better control over the batch running. You can specify what processes need to run in sequence and what process needs to wait for other processes to complete.

```
WAITFOR _ALL_ SASJOB1 SASJOB2;
```

SYSTASK along with list option is used to list the currently active job(s) and along with kill option can be used to kill a job.

```
SYSTASK LIST _ALL_;
SYSTASK KILL SASJOB;
```

Regarding performing other activities using SYSTASK the commands will be similar to what is defined in the X

statement of the above table.

Options associated with each windows command is detailed in the following table.

| COMMAND  | OPTIONS | PURPOSE   |
|----------|---------|---|
| MKDIR    |         | To Create a directory   |
| RMDIR    | /S      | To remove the directory tree which includes sub-directories and files   |
|          | /Q      | Does the same activity as /S option but without giving a prompt to delete the directory tree  |
| DIR      | /A:D    | Displays the attributes of directories in that directory, /A :-D, excludes directories and displays other information   |
|          | /A:R    | Displays attributes of the read only files  |
|          | /A:S    | Displays attributes of system files.  |
|          | /B      | Prints the bare format and no heading information.  |
|          | /Q      | Displays the owner of the file  |
|          | /S      | Displays the files in the specified directory along with the subdirectories   |
|          | /T:C    | Date & time of creation of the file   |
|          | /T:A    | Date & time of last access  |
|          | /T:W    | Date & time of last written/updated   |
| COPY     | /V      | Verifies that new files are written correctly   |
|          | /Y      | Suppress the prompting to overwrite the destination file  |
|          | /A      | Indicates an ASCII text file  |
| XCOPY    | /S      | Copies subdirectories and empty ones  |
|          | /V      | Verifies the size of the each new files   |
|          | /C      | Continues copying even if the error occurs  |
|          | /Q      | Doesn't display the filename while copying.   |
|          | /R      | Overwrite read only files.  |
|          | /U      | Copies only files that are already in the destination directory   |
|          | /K      | Copies all the attributes along with the files  |
|          | /Y      | Suppress the prompting to overwrite the destination file  |
| ATTRIB   | +R      | Sets read only file attributes, -R removes the read only file attributes  |
|          | +A      | Archive file attributes, -A removes the archive file attributes   |
| MOVE     | /Y      | Suppress the prompting while moving files from one directory to other (MOVE command is also used for renaming the existing directory to other name)                               |
| DEL      | /P      | Prompts for confirmation before deleting the file   |
|          | /F      | Force to delete the read-only files   |
|          | /S      | Deletes the specified files from sub-directories as well  |
|          | /Q      | Deletes the files without prompting   |
| ROBOCOPY |         | This is called as robust copy which has wide range of activities you can you. You can find information regarding this when you specify "HELP ROBOCOPY" in the CMD command window. |

## HOST: UNIX OPERATING SYSTEM

| X STATEMENT   | PIPE DEVICE TYPE  |
|---|---|
| <p>***1) To Create a Directory for copying the existing set of outputs;</p> <pre>X "mkdir /my/unix/newdir";</pre> | <p>***1) To Create a Directory for copying the existing set of outputs;</p> <pre>filename makeinfo pipe "mkdir /pharmsug/unix/newdir"; data _null_; infile makeinfo; run;</pre> <p>Note: Use data _null_ statements as above to execute any filename statement.</p> |
| <p>***2) Copying to the directory;</p> <pre>X "cp /pharmsug/unix/*.rtf /pharmsug/unix/newdir/.";</pre>            | <p>***2) Copying to the directory;</p> <pre>filename copyfl pipe "cp /pharmsug/unix/*.rtf /pharmsug/unix/newdir/.";</pre>   |
| <p>***If You want to delete the existing files in the directory;</p> <pre>X "rm /pharmsug/unix/*.rtf";</pre>      | <p>***If You want to delete the existing files in the directory;</p>  |
| <p>***3) To Change the attributes of the copied files in the archived folder;</p> <pre>X "chmod ugo-wx</pre>      |   |

```

/pharmsug/unix/newdir/*.rtf" ;
filename delfl pipe "rm
/pharmsug/unix/*.rtf" ;

***4) Now Fetch the file attributes;

X "cd /pharmsug/unix/newdir; ls -l *.rtf >
fileinfo.txt";

***To fetch the owner of the file;

X " ls -al *.rtf | awk '{print $3}';

***5) Reading the file attributes into a dataset;

data outf;
  infile "fileinfo.txt" dlm=' ';
  length filenm $50 permisn $10;
  input permisn $ nl $ user $ group $
  size $ month $ day $ year $ filenm $ ;
  filedate=
    input(compress(day||month||year),
          date9.);
  format filedate date9.;
run;

6) Command to send e-mail with the content of the file
logcheck.txt in the body.
X "(cat logcheck.txt;) | mailx -s "Log
file from last run" toperson@x.com -c
ccperson@x.com

filename delfl pipe "rm
/pharmsug/unix/*.rtf" ;

***3) To Change the attributes of the copied files in the
archived folder;

filename permi pipe "chmod ugo-wx
/pharmsug/unix/newdir/*.rtf" ;

***4) Now Fetch the file attributes;

filename fileinfo pipe 'ls -l
/pharmsug/unix/newdir/*.rtf ' ;

***To fetch the owner of the files;

filename owner pipe " ls -al *.rtf | awk '{print $3}";

***5) Reading the file attributes into a dataset;

data outf;
  infile fileinfo dlm=' ';
  length filenm $80 permisn $10;
  input permisn $ nl $ user $ group $
  size $ month $ day $
  year $ filenm $ ;
  filedate=
    input(compress(day||month||year),
          date9.);
  format filedate date9.;
run;

```

\*\*\*6) Create CSV file using filename, Create blank files;

```

X 'touch newfile.csv';
filename copyin pipe "ls -l";
filename copyout "file.csv";
data _null_;
  infile copyin ;
  file copyout;
  input;
  put _infile_;
run;

```

**SYSTASK (UNIX)**

SYSTASK for UNIX operating system is similar to the windows. The shell option invokes the command with the operating system environment. It helps to process the special characters and the aliases mentioned within the parenthesis.

1) Simple list UNIX command.  
 Systask command " ls -l \*.lst " taskname=tsklist status=lststat shell;

2) Grep command that outputs to a txt file.  
 Systask command "grep ""TABNO"" \*.lst > check.txt" taskname=tshsrch status=srchst wait shell;

Options associated with each UNIX command is detailed in the following table.

| COMMAND | OPTIONS | PURPOSE  |
|---------|---------|--|
| MKDIR   |         | To Create a directory  |
| RMDIR   |         | Removes empty directory  |
|         | -r      | To remove the directory tree which includes sub-directories and files within                               |
|         | -f      | Does the same activity as -r option but also deleted write protected files as well.                        |
|         | -i      | Interactive mode asking for confirmation.  |
| LS      | -l      | List and displays the attributes of files and directories in the current directory alphabetically by name. |

| COMMAND | OPTIONS | PURPOSE  |
|---------|---------|--|
|         | -r      | Same as -l option with reverse order.  |
|         | -t      | Sorts by timestamp, latest first   |
|         | -u      | Uses last accessed time stamp instead of last modified.                              |
|         | -a      | List all entries including the hidden files ones that start with .(dot).             |
| CP      |         | Copies files or directories  |
|         | -p      | Preserves the attributes like timestamp, owner, etc.                                 |
|         | -i      | Interactive mode   |
|         | -r      | Copies directories and its content and all subdirectories and their content as well. |
| CHMOD   |         | Sets file attributes.  |
| RM      | -r      | Removes directories and the files within   |
|         | -i      | Prompts for confirmation   |
|         | -f      | Deletes files without prompting  |
| DIFF    |         | Will compare the two files and print out the differences between                     |
|         | -r      | Will compare contents of 2 directories looking for similar files.                    |
|         | -i      | Ignores case of letters  |
|         | -b      | Ignores trailing blanks  |
|         | -w      | Ignores all blanks   |
| CHOWN   |         | Change ownership of a file or directory to one or more users                         |

### EXAMPLE: AUTOMATE GENERATION OF DATA LISTINGS

For suppose consider data management listings that are generated for the ongoing clinical study every month once the ETL is done from the database management systems like ORACLE Clinical® or Datalabs® or any other database system. The ETL is scheduled to takes place every 15<sup>th</sup> of every month and the programmer generates the data management listings once the refresh of the data is done and send them to data management team so that they can review the data. This process can be automated utilizing the operating system commands and SAS coding which will reduce the programmer's involvement and can also reduce the man made errors once the whole process is setup for that activity.

### SCHEDULING THE TASK IN WINDOWS

Setting up the windows task scheduler for running the batch program periodically, to run it every day once the user login you can do that by using the command line or windows task scheduler which is a GUI tool.

AT command: Scheduling of the tasks can be done by using the AT command at command prompt in CMD window.

AT 10:49 /EVERY:m, t ,w, th, f c:\pharmsug\utilities\trigger.bat (you can find more about this in windows help).

GUI Scheduler: Schedule of jobs can also be done by using the windows scheduler tool

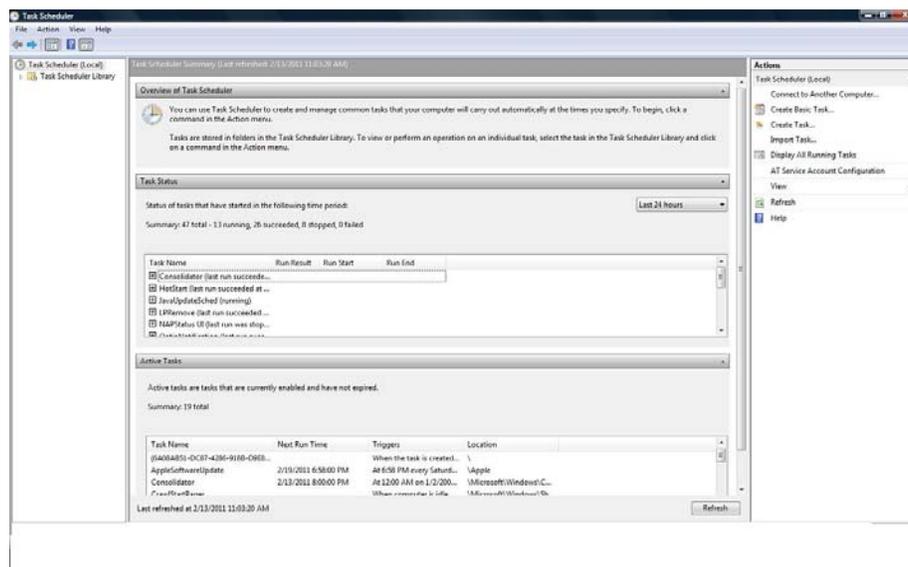


Figure 1: Task Scheduler under Windows Vista.

**FLOWCHART SHOWING THE PROCESS AUTOMATION**

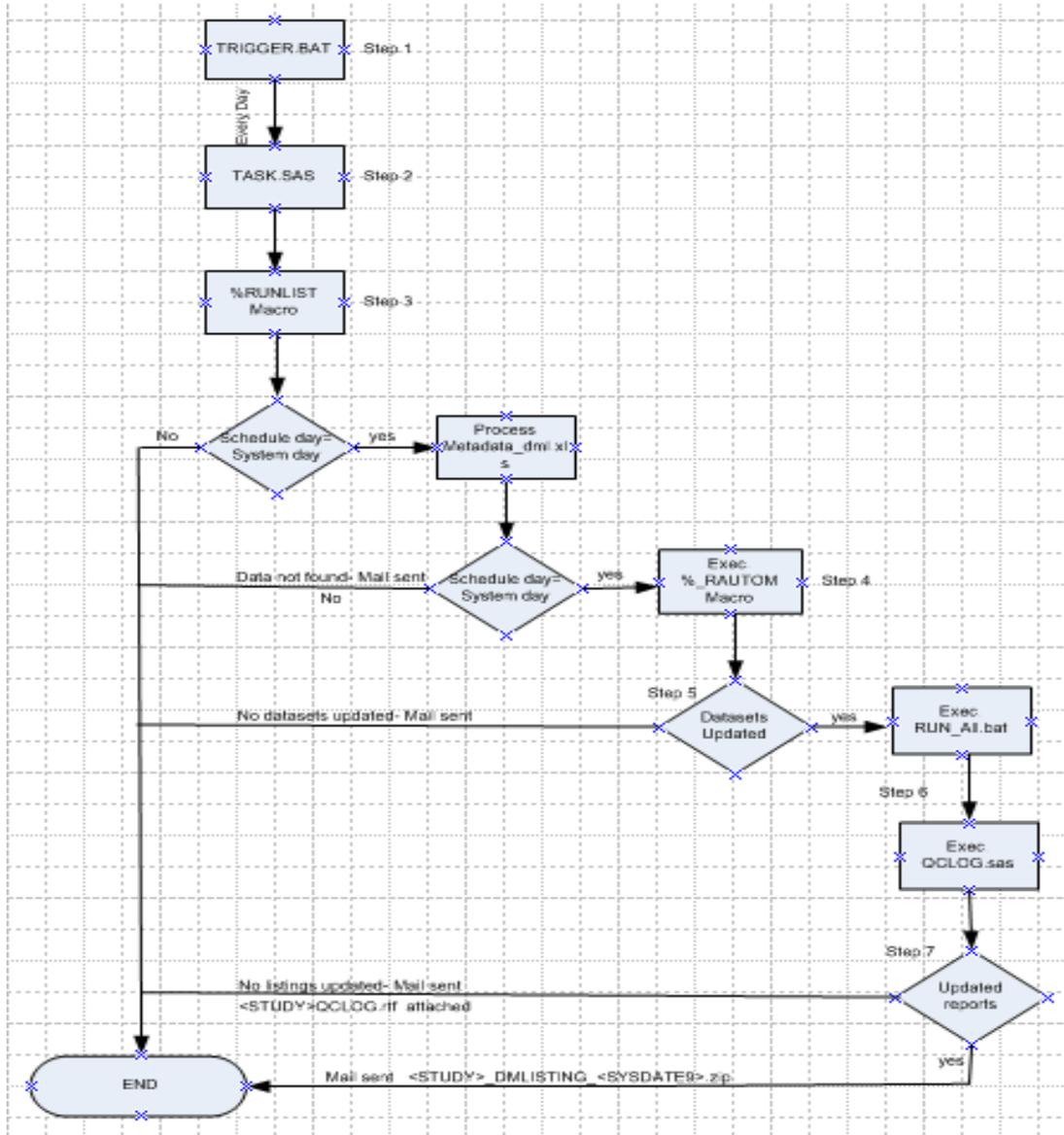


Figure 2: Process Flowchart Showing Each Step of Automation

**STEP1: TRIGGER.BAT**

The batch program can be written to schedule to run different jobs on different dates, so the file TRIGGER.BAT will have following code calling different programs that you want to run on different days. The below example explains how to set up the batch run for two batch programs.

If we need a batch program that executes two batch files, one batch file to execute on day1 and another batch file day2 of the month, it can be coded as the below example shows.

```

@echo off
for /F "usebackq tokens=1 delims=/ " %%i
in ('%DATE%') do set dayofthemonth=%%i

IF /I %day%==17 goto batch1
IF /I %day%==20 goto batch2

goto finish

:batch1
"C:\pharmsug\test\test1.bat" (Continued..)

:batch2
"C:\pharmsug\test\test2.bat"
echo execute batch2

:FINISH
echo execute batch1
GOTO FINISH
  
```

But in this scenario there is only need for one batch program TRIGGER.BAT that needs to be scheduled to execute daily so we directly schedule this batch program through windows scheduler. This batch program executes "TASK.SAS" program which calls the macro "%RUNLIST" that is designed to execute 5 different tasks on different dates. The batch program is set to run every morning 10:49 am and only when the system day is equal to the day of the month specified for a particular task then reports/listings are generated for that task defined in the TASK.SAS program.

The batch program TRIGGER.BAT will be written as

```
"C:\Program Files\SAS\SAS 9.1\sas.exe" -sysin TASK.SAS *if the batch program is in the same location as of TASK.SAS program;
```

Or else the program can be written as "start /wait sas C:\ utilities\task\task.sas".

**STEP2: TASK.SAS**

This program will call the %RUNLIST macro and the parameters are passed to that macro. For scheduling only one task the %RUNLIST macro parameters are passed accordingly.

```
%RUNLIST(TASK1=%str(C:\pharmsug\testlisting), DAY1=10, LABEL1=%str(This is a test run), NTASK=1);
```

**STEP3: %RUNLIST MACRO**

```
%RUNLIST(TASK1=, DAY1=, LABEL1=, TASK2=, LABEL=2, DAY2=, TASK3=, LAEBL3=, DAY3=, TASK4=, LABEL4=, DAY4=, TASK5=, DAY5=, LABEL5=, NTASK=)
```

The %RUNLIST macro is designed to run five different tasks on five different dates

Parameters TASK1-TASK5 has to be passed with path where the METADATA\_DML.xls excel file for each task is located

Parameters DAY1-DAY5 should be passed with information on which day of the month each task should be executed.

Parameters LABEL1-LABEL5 are for labeling each task and NTASK= is for letting the system know how many number of tasks are present and the maximum would be 5.

The METADATA\_DML.xls file will be having two columns Category and Information, so the user needs to pass on the correct information for each category, below is the table how the information should be passed.

| CATEGORY  | INFORMATION   |
|---|---|
| Protocol Name:  | Enter Study name  |
| Path of Datasets:                                       | Enter the path where datasets are present   |
| Path of Programs for Listings:                          | Enter where the programs for listings are present   |
| Path of Utility Macros:                                 | Enter where the utility macro programs are located, some QC programs  |
| Path of _RAUTOM Macro Program:                          | Enter the path of the program where _RAUTOM is located  |
| Is Latest Data Extract is Correct: (Y or Blank)         | Even if the data doesn't gets updated if the program needs to generate listings then Y else blank                     |
| Is the latest Listings Refresh is Correct: (Y or Blank) | Even if the listings doesn't gets updated if the program needs send those listings to mailing group then Y else blank |

If all the required information is passed in the excel sheet then the program executes %\_RAUTOM macro is executed, else an e-mail is send to the user to check for the issues in the information entered in the excel workbook.

#### **STEP4: %\_RAUTOM MACRO**

```
%_RAUTOM(STUDY=, DSPATH=, PGPATH=, UTLPATH=, DSRIGHT=, LSTRIGHT=)
```

This macro generates listings/reports for a particular study, based on the scheduled day and zips the listings and sends them to the mailing list after the listings/outputs are generated.

The parameter values for this macro are passed based on the information entered in the excel file, METADATA\_DML.xls.

#### **DESCRIPTION OF PARAMETERS**

STUDY: Study name.

DSPATH: Path of the datasets location.

PGPATH: Path where the programs and outputs are located.

UTLPATH: Path where utility macros are located.

DSRIGHT: If the user needs the outputs irrespective of whether the datasets are updated or not then DSRIGHT is set to 'Y'.

LSRIGHT: If the user needs the e-mail to be send with the outputs irrespective of all the listings are updated or not then LSTRIGHT is set to 'Y' in the excel file.

#### **STEP5: CHECKING FOR UPDATES IN THE DATASETS**

The %\_RAUTOM macro first checks if all the datasets in the specified library are updated, i.e. by first deleting the previous modification information text file named "lstmodified.txt" (which is previous modification dates information of previous run) and rename the "latest.txt" (latest modification dates information of previous run) to "lstmodified.txt" and generates latest.txt having the latest dates of the updated datasets and then compares with dates with previous dates. If all the datasets are updates then program will executes "RUN\_ALL.BAT" program which is located in the directory where the listings programs are present to generates the listings, else it will stop executing and will send an e-mail to the user saying the task was ABEND and the user can cross check and execute manually if he needs to generate them.

#### **STEP6: SCANNING THE LOGS OF THE LISTINGS PROGRAMS**

The %\_RAUTOM macro then executes the QCLOG.SAS utility macro to scan the logs of the executed listing programs and it generates an RTF file named as <study>\_qclog.rtf, which will have all the ERROR, WARNINGS information of the logs.

#### **STEP7: SENDING EMAILS TO MAILING LIST**

After the QCLOG.SAS program executed, the %\_RAUTOM macro then cross check if all the listings are updated on the specified day. If all the listings are updated then it zips all the listings to a zip file and name it as <study>\_dmlistings\_<sysdate9>.zip and sends e-mail to the mailing list attaching the zip file, else if any of the listings are not updated then it sends a mail to the user attaching the QCLOG.SAS generated RTF file asking to investigate the reason for not updating.

#### **COMPARISON BETWEEN SYSTASK, X, FILENAME WITH PIPE**

We can give multiple commands in single X statement unlike Filename options which only allows one UNIX command in single filename statement.

Filename option with pipe device always creates temporary output that can be used later in the programming while with X statement we can create permanent output files only.

SYSTASK command by default executes asynchronously while the X command is synchronous. With the options XWAIT/ NOXWAIT, XSYNC/ NOXSYNC for X command and the WAIT/ NOWAIT and WAITFOR for SYSTASK we can control the way the commands execute.

With SYSTASK you can execute multiple statements simultaneously.

SYSTASK with various other options like TASKNAME and STATUS also provides better control and flexibility.

CALL SYSTEM and %SYSEXEC can also be used to execute Operating System commands synchronously similar to X. CALL SYSTEM statement can be executed within a data step. XWAIT/ NOXWAIT and, XSYNC/ NOXSYNC options can be used with these as well.

## **CONCLUSION**

Even though there are multiple ways to execute operating system commands using SAS it is up to the programmers to decide which way works best for them. Each way has its own set of pros and cons.

## **REFERENCES**

- 1) Na Li, Applications for Running DOS Commands within SAS, Paper PO13.
- 2) Windows Help Documentation for DOS Commands.
- 3) UNIX Help Documentation for UNIX Commands.

## **ACKNOWLEDGMENTS**

We would like to thank Patricia Rice, Sr. Director CliniRX Research (USA) for proof reading the draft document of the paper and support.

## **CONTACT INFORMATION**

Your comments and questions are valued and encouraged.

### **CONTACT DETAILS:**

Ranganath Bandi

CliniRX Research (USA)

Chicago, IL

rbandi@clinirxusa.com;

ranganathb@evantagetechologies.com

Harini Kunduru

Bristol Myers Squibb Company

Pennington, NJ

harini.kunduru@bms.com; harnikunduru@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.