PharmaSUG 2021 - Paper AP-132

100% Menu Driven SAS Windows Batch Job Runner and Log Analyzer

Sathish Saravanan and Kameswari Pindiprolu Quartesian Research Private Limited, Bangalore, India;

ABSTRACT

SAS is the widely used software in clinical research industries for clinical data management, analysis & reporting. Every clinical study requires numerous SAS programs to be developed for generating Datasets (D), Tables (T), Listings (L) and Graphs (G). The program rerunning is evident due to programming specification changes for DTLG and Ongoing Studies. At the same time for each rerun, it is mandatory to ensure the logs of every submitted program is free from SAS defined potential messages. Various methods have been discussed about bulk rerunning of programs and analysing the log and none of them is menu driven approach. Companies are investing so much of money to develop and maintain such a menu driven tool that performs bulk rerunning of the SAS programs along with the log analysis. The purpose of this paper is to provide a Microsoft Excel VBA based tool that is 100% Menu Driven or Graphical User Interface to perform bulk running of the SAS programs along with Log analysis, process summary and easy access to the particular log file from the several files. It also creates a SAS program file with all the selected programs to be used for the later use. The proposed approach needs only 4 inputs from the user as follows.

1. Program Location
2. Log location for the log files to be saved.
3. LST output location for the LST output file to be saved.
4. Selection of the programs to run.

INTRODUCTION

For the clinical research programmers working in an CRO or a pharmaceutical company the crux in the difficulty of the job lies in dealing with the complex data. The complex nature of the data comes because the clinical trials are done drugs targeting numerous diseases. The list of these diseases is becoming exhaustive, and the drugs needed to cure or manage them is becoming extensive. When the programmers are treading the fine line between quality vs time in these fast-paced turbulent times the next major challenge comes with the redundancy of the work.

In the primary stage of the development of the programs the programmer spends copious amounts of time on building the datasets and generating the Tables, listings, and Graphs. As the programming work is mostly started either with the test data or with minimal data like the data captured during the first patient first visit (FPFV) the programmer tries to write the programs in a very robust manner anticipating that the programs would be able to handle any all kinds of issues. During this phase, the programmer carefully checks all the programs individually for its completeness, robustness, logics, and log issues. Once the clinical trials begin to recruit more subjects into the study the programmers are flooded with new data on a periodic basis. During this phase, the redundancy of the work burdens the programmers.

In the initial stages of programming the programmer is bound to run each program individually to check for the log issues and the completeness in generating the right outputs and the workload and the tediousness is non-negotiable. But once this phase is surpassed and the new data transfers keep coming to the programmers then the redundancy and the repetitiveness in the work begins where the programmers must run all the programs time and again and check the logs and outputs individually.

To tackle this repetitiveness of the work we need procedures that would reduce the manual rerunning of the programs and log checks.

BATCH RUNS

One of the approaches that is followed currently in the industry to overcome this repetitive work is running SAS programs in batch mode. To run the programs in batch mode we need to create .BAT file. Which is created by opening a notepad and listing all the programs that are required to be run sequentially and saving it with .BAT extension. 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. Once the .BAT file is created it is run in the command prompt. After all the programs are run, we can manually check the log files for any log issues and then go back to update the programs if required. Though this approach has its advantage over the individual program rerun but still it would still pose certain hinderances like we need to list all the programs manually. Furthermore, you need to make sure that you have spelled the names of the programs correctly and entered the correct path. This can be quite challenging as depending on the size of the study, there are sometimes dozens or even hundreds of programs that need to be listed in the correct order.

We have come up with another approach that would make the Batch runs even more easy. A SAS Batch Job Runner.xlsm has been created using the VBA in excel. This SAS Batch Runner has rectified many of the difficulties faced that are faced by the programmer while doing the batch runs using .BAT file in command prompt.

VISUAL BASIC APPLICATIONS

VBA is a combination of the Microsoft's event-driven programming language Visual Basic with Microsoft Office Applications such as Microsoft Excel. VBA allows user to customize beyond what is normally available with MS Office host applications and by manipulating the graphical-user-interphase (GUI) features such as toolbars, menus, dialogue boxes, and forms to create user defined functions (UDF`s) and automate the required processes and calculations. As it is an event driven tool, which means that you can use it to tell the computer to initiate an action or string of actions. For doing this we need to build a custom VBA code.

Customized macros have been written to macrotize the various functions of the SAS Batch Job Runner and Log Analyzer. In order to make it a Graphical User interface these macros have been linked to custom buttons.

VBA can work in external-that is, non-Microsoft settings by using a technology called COM interface, which allows the command to interact across computer boundaries.

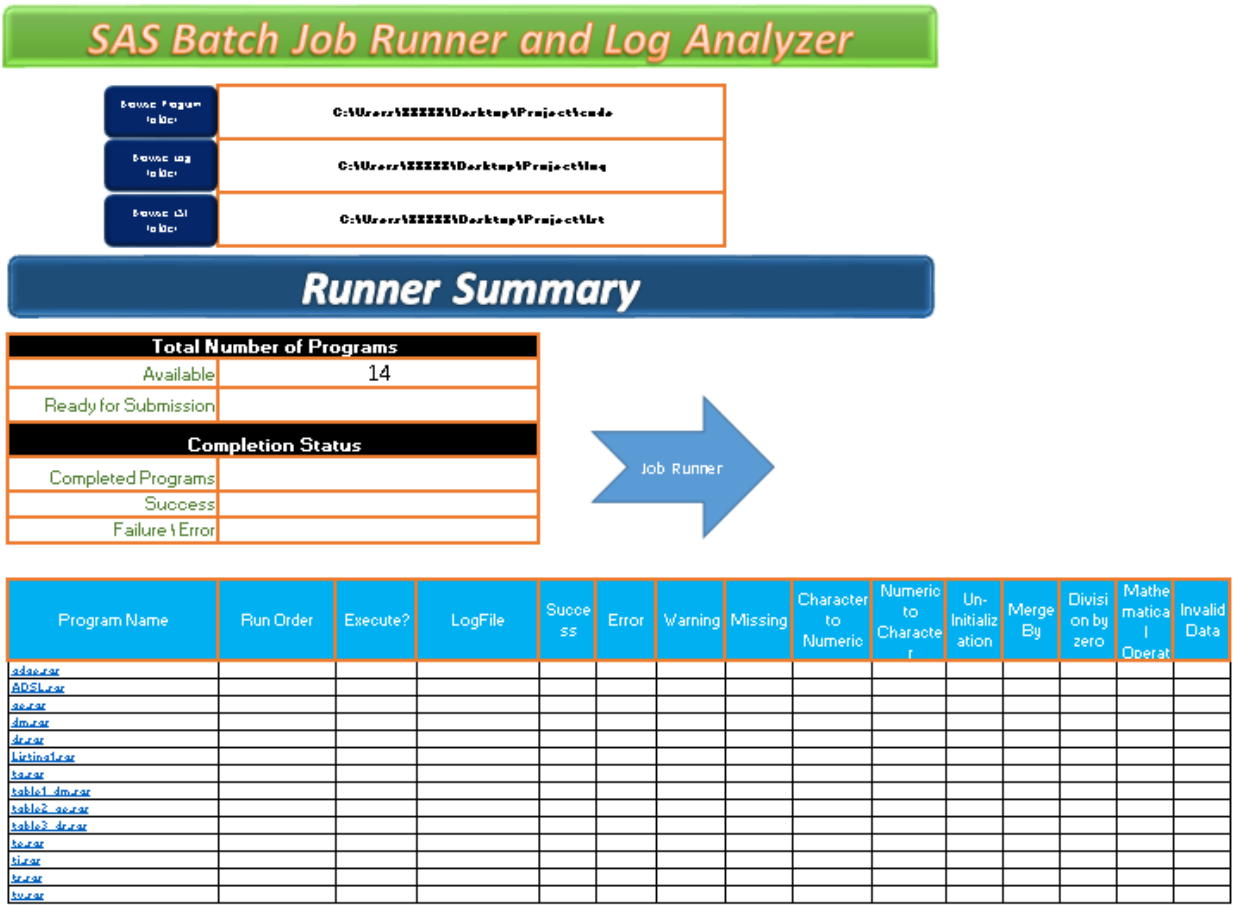
SAS JOB RUNNER

Using Visual Basic Applications, we have created a SAS Batch Job Runner which will help in the batch running of codes, also having a host of advantages than using the traditional .BAT files reruns.

Below given are the inputs that are required for running the SAS Batch Job Runner. Also, it must be run in an environment that has SAS installed in it.

The proposed approach needs only 4 inputs from the user as follows.

1. Program Location
2. Log location for the log files to be saved.
3. LST output location for the LST output file to be saved.
4. Selection of the programs to run.



Display 1. SAS Job Runner & Log Analyzer Interface.

The first Custom button that is created in the SAS Batch Job Runner and Log Analyzer is the “Browse Program Folder”. When the user clicks this button it browses for the program folder which would contain all the SAS codes that the user intend to rerun. Then this macro would also give the total number of programs in the code folder which can be viewed in the “Available” row of the “Runner Summary”. The list of the individual programs can be seen under the program name column.

Display 2. Custom Buttons for selecting programs, Logs & LST.

Runner Summary

Total Number of Programs	
Avilable	14
Ready for Submission	0

Display 3. Total Number of Available programs

Program Name	Run Order	Execute?	LogFile	Success	Error	Warning	Missing	Character to Numeric	Numeric to Character	Un-Initialization	Merge By	Division by zero	Mathematical Operations	Lost Card
adae.sas														
ADSL.sas														
ae.sas														
dm.sas														
ds.sas														
Listing1.sas														
ta.sas														
table1_dm.sas														
table2_ae.sas														
table3_ds.sas														
te.sas														
ti.sas														
ts.sas														
tv.sas														

Display 4. Individual Listing of programs

Once the programs are listed then the user can click the “Browse Log Folder” custom button which would select the log folder if its already created, or it would create a new log folder if there isn’t any, to store the log files. The Next Button “Browse LST folder” would also similarly select the LST folder created to store the created Tables, Listings and Graphs or create a folder if it is not present. All the custom buttons are named in a way they are self-explanatory and can be used by a very less experienced person as well.

The we need to define the order in which we need to run the SAS programs in the “Run Order” column. This is important as in the code folder the codes can be saved in any order, but for the programs to run properly based on the dependency we need to properly give the order in which the codes should run. This can be done simply by attributing the programs with numbers in which we want them to run ascendingly.

Program Name	Run Order	Execute?
adae.sas		
ADSL.sas		
ae.sas	6	0
dm.sas	7	1
ds.sas	8	1
Listing1.sas		
ta.sas	1	1
table1_dm.sas		
table2_ae.sas		
table3_ds.sas		
te.sas	2	1
ti.sas	3	1
ts.sas	4	1
tv.sas	5	1

Display 5. Run Order & Execute

There might be many instances in the reruns where we may not want to rerun all the programs that have a run order. To tackle that issue we have created a “Execute?” column which takes binary values of 0 and 1. So which ever programs we want to run will be given “1” and the others a “0”. This feature will further make the tool even more efficient and the reruns customizable as per the programmer’s requirement. Suppose if do not give the run order for any program but still give a “1” for the “Execute?” then that specific program would run after all the programs have been rerun.

Once these inputs are given, we can go ahead and click on the “Job Runner” custom button. Another major requirement that this job runner would require is having the SAS installed in the same environment that the job runner is placed. Once the “Job Runner” button is clicked then it would go and invoke the SAS session and then run the programs as numbered in the run order column and as per the “yes/no” from the execute column.

After the rerun is executed by the Job runner. The number of programs that have been rerun is given in the “Completed Programs” row. This number is further bifurcated into number of programs that have been successfully rerun without error or any execution failure into the “Success” row and the rest into the “Failure \ Error” row.

Total Number of Programs	
Avilable	14
Ready for Submission	7
Completion Status	
Completed Programs	7
Success	5
Failure \ Error	2

Display 6. Run Order & Execute

After the reruns, the log files that are generated can be seen in the “Log File” column along with the location of the individual log files which are hyper linked. The next column “Success” is coded to give “Yes” for the programs that have been successfully rerun without any log issues. If there are any log issues produced by the program during the rerun, then it is indicated by “No” in the “Success” column.

To track what type of log issues are generated during the non-successful run of the program, the Log analyzer has been macrotized to capture the commonly encountered log issues like “Errors”, “Warning”, “Missing”, “Character to Numeric”, “Numeric to Character”, “Un-Initialization”, “Merge By”, “Division by zero”, “Mathematical Operations” and “Invalid Data”. If any of these log issues are generated by the program, then they are denoted in the specific column by the number of the specific issues that are encountered.

Program Name	Run Order	Execute?	LogFile	Success	Error	Warning	Missing	Character to Numeric	Numeric to Character	Un-Initialization	Merge By	Division by zero	Mathematical Operations	Invalid Data
adae.sas														
ADSL.sas														
ae.sas	6	0												
dm.sas	7	1	dm.log	Yes	0	0	0	0	0	0	0	0	0	0
ds.sas	8	1	ds.log	Yes	0	0	0	0	0	0	0	0	0	0
Listing1.sas														
ta.sas	1	1	ta.log	Yes	0	0	0	0	0	0	0	0	0	0
table1_dm.sas														
table2_ae.sas														
table3_ds.sas														
te.sas	2	1	te.log	Yes	0	0	0	0	0	0	0	0	0	0
ti.sas	3	1	ti.log	No	0	1	0	0	0	0	0	0	0	0
ts.sas	4	1	ts.log	Yes	0	0	0	0	0	0	0	0	0	0
tv.sas	5	1	tv.log	Yes	0	0	0	0	0	0	0	0	0	0

Display 7. Final Report

Once this report is generated in the GU interface of the tool then it becomes very easy to identify the log issues with a glance. Then the programmer can directly click on the hyperlink of the log file which would directly take the user to the specific log location and the programmer can easily identify the issue and then come back to the SAS Job runner and click the specific program which are also hyperlinked which would then open the program in a SAS editor window where the program can be edited and saved.

CONCLUSION

This SAS Batch Job Runner and Log Analyzer hosts a plethora of advantages than the other forms of batch reruns like first and foremost we can customize the rerun as per our requirements and we can get a complete rerun report. It also gives the number of successful programs along with the status of the rerun for the individual programs. When a log issue is identified from the list of issues that are tracked in the interface, we can click on the hyperlink of the log files and the programs and then open them in SAS session and update them directly. This would reduce an abundance of load on the programmer where in the traditional batch runs where even though we can have log analyzers placed still we need to go and open the log files and programs physically and then update them. This feature would act as a serious time saver. Also, finally as it is a graphical user interface it makes it very easy to use it and would play a huge role in delegation of work to other less experienced programmers also.

All these things make this SAS Batch Job Runner and Log Analyzer an idea worth to put to actual use and also explore further for its even more finer refinement.

CONTACT INFORMATION

Sathish Saravanan
Quartesian Research Private Limited, Bangalore, India.
sathish.saravanan@quartesian.com

Kameswari Pindiprolu
Quartesian Research Private Limited, Bangalore, India.
Kameswari.p@quartesian.com