

SmartLog - A UNIX based Utility

Satish Honrao, Syneos Health, Pune, MH, India

ABSTRACT

Clinical trial sponsors are seeking regulatory approvals from authorities like USFDA and EMEA. In order to accomplish this, Sponsors must submit detailed clinical study reports (CSRs) and individual participant data. Drug companies would like to do this with minimum possible time without compromising the quality. For CSR submissions, we have to submit the SAS® codes, logs and listings of the generated clinical tables, listings and graphs, to the regulatory authorities. This paper will explore automated approach to submit SAS codes and collating all the issues found in the log.

INTRODUCTION

During the eleventh hour of submission, one can end up in a state of commotion as there might be hundreds of SAS codes to run, then checking each log file for accuracy. If SAS system is Window-based, then it is somewhat easier to track the issues due to its friendly GUIs. But, it is really cumbersome job at the submission time; especially in UNIX® based SAS.

To execute this facility in UNIX, there are some prerequisites:-

- 1] All the completed SAS codes should be present in the program directory.
- 2] Program owner should have a read/write/execute access in the given UNIX directory.
- 3] User should be able to launch SAS from his/her UNIX terminal.

Functions of this facility are:-

- 1] Running all SAS codes in one go.
- 2] Identifying any errors/warning/unwanted messages in the log.
- 3] Collating all the issues of #2 in a single file.

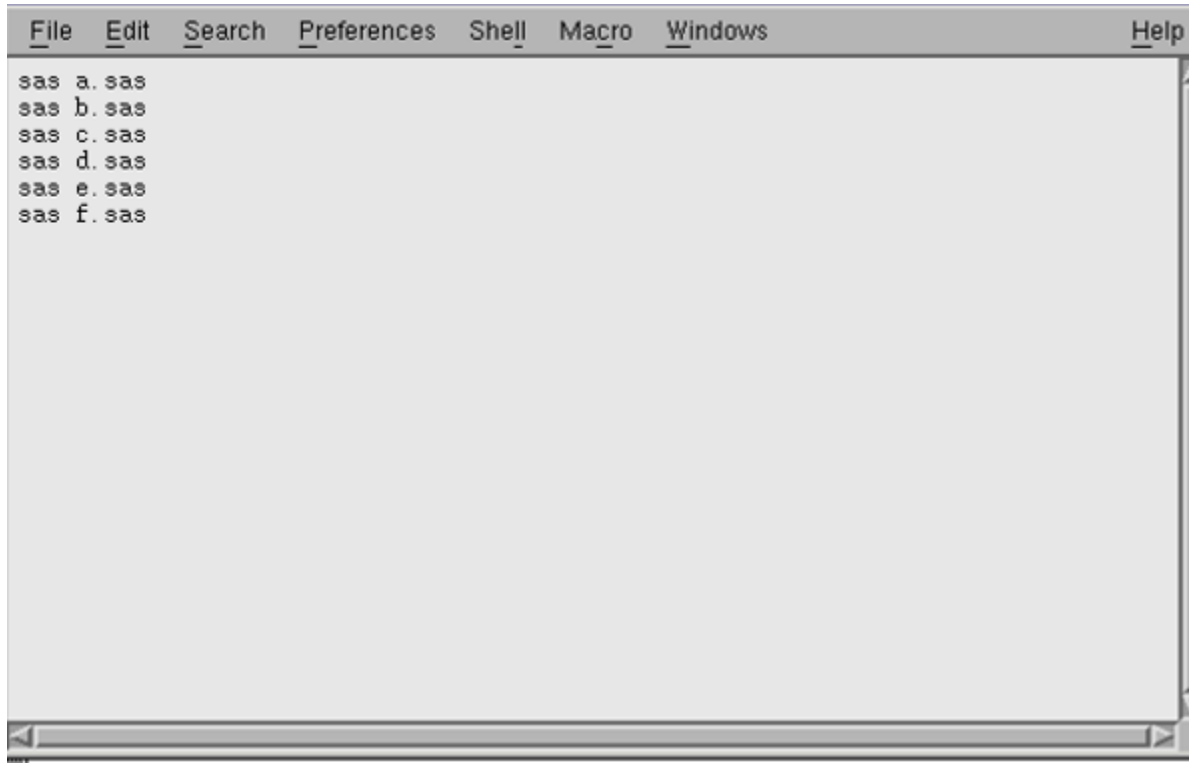
PROCESS OVERVIEW

Running all SAS codes:

To elaborate this, shell has been used which will have the list of all SAS programs present in a directory.

Although there are many ways to run multiple SAS codes in UNIX, the simplest of them all is to just include all .sas program names in a file one by one. Before that, please make sure that all SAS Programs are present in a directory and working.

This file can be created in following way:

A screenshot of a text editor window with a menu bar at the top containing 'File', 'Edit', 'Search', 'Preferences', 'Shell', 'Macro', 'Windows', and 'Help'. The main text area contains six lines of text, each starting with 'sas' followed by a space and a filename: 'sas a. sas', 'sas b. sas', 'sas c. sas', 'sas d. sas', 'sas e. sas', and 'sas f. sas'. The window has a standard scroll bar on the right side.

After saving this file, the access permission of this file is changed to 755 to make it executable from UNIX prompt. This file can be executed either by “sh filename” command or simply prefixing “./” before filename. Through this command, we can run multiple files in one go.

As we run SAS program in UNIX, there are two files namely LOG and LST get created for each and every programs.

Finding errors and warnings:

Using combination of UNIX command in shell, we can find out errors, warnings and unwanted messages. This is really useful in debugging purposes. By adding different wildcards in GREP command in shell, we can singled out all the unwanted messages.

Following is the script about the same:

```
File Edit Search Preferences Shell Macro Windows
#!/bin/sh
#!/bin/sh

# Check SAS log files for errors, warnings and other unwanted messages

CHK='basename $0'

#echo ""

case $# in
  0)
    echo -e "-----"
    echo "$CHK: Check SAS log files for errors, warnings and other unwanted messages "
    echo "Usage: $CHK [FILE]"
    echo "  For example, $CHK *.log will check all log files in the current directory."
    echo "                $CHK mysas.log will just check the mysas.log file."
    echo "-----"
    echo ""
  ;;
  *)
    for fname in $*
    do
      grep -E -H -n -i "warning|error|uninit| 0 obse|no obser|repeats of|not found\
|not valid|invalid|syntax error|overwritten|converted|missing values were generated\
|outside the axis range|duplicate by var|not created|contained a missing|ignored\
|went to new line|no effect|division by zero|due to loop" ${fname}
    done
  ;;
esac
```

To run this script, we have to make this file executable by changing its access permission mode to 755. We can run this script either as “scriptame *.log” or “scriptname progame.log”.

Collating all the issues in a single file:

After the running the script, we can see the results on the command line itself or we can re-direct the results to another temp file. Later on, this temp file can be viewed using any available editor in UNIX.

Let's say the filename of script is “loglst”. Then, we can invoke the script either of the following:

- 1] loglst *.log
- 2] loglst *.log > temp.txt

This is a screenshot of results:

```

      /temp1$ loglst *.log
a.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
b.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
c.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
d.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
d.log:51:ERROR 22-322: Syntax error, expecting one of the following: !, !!, &, *, **, +, -, /, <, <=, <>, =, >, ><, >=, AND, EQ, GE, GT, IN,
d.log:56:NOTE: The SAS System stopped processing this step because of errors.
d.log:57:NOTE: SAS set option OBS=0 and will continue to check statements. This might cause NOTE: No observations in data set.
d.log:58:WARNING: The data set WORK.DD may be incomplete. When this step was stopped there were 0 observations and 4 variables.
d.log:75:ERROR: Errors printed on page 1.
      /temp1$ █

```

If the above result is stored in another file, we can view the file using any available editor in UNIX and check the results. Using this method, command prompt always remains free and can be utilized to do other tasks.

The screenshot of redirected output file is as follow:

```

File Edit Search Preferences Shell Macro Windows
a.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
b.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
c.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
d.log:22:You are running SAS 9. Some SAS 8 files will be automatically converted
d.log:51:ERROR 22-322: Syntax error, expecting one of the following: !, !!, &, *, **, +, -, /, <, <=, <>, =, >, ><, >=, AND, EQ, GE, GT, IN,
d.log:56:NOTE: The SAS System stopped processing this step because of errors.
d.log:57:NOTE: SAS set option OBS=0 and will continue to check statements. This might cause NOTE: No observations in data set.
d.log:58:WARNING: The data set WORK.DD may be incomplete. When this step was stopped there were 0 observations and 4 variables.
d.log:75:ERROR: Errors printed on page 1.

```

If you see the above outcome, we can concur that all the errors, warnings and unwanted messages (like repeats of by, uninitialized, syntax error) are trapped from all the logs present in that particular folder.

CONCLUSION

The ultimate goal of this paper is to provide a method of job automation at the time of running multiple SAS codes simultaneously and keep a track of all issues arisen thereafter. At the time of submission or even if one just want to run the bulk of programs at one go, this script is convenient to use, implement and infer the results from it.

REFERENCES

UNIX reference: <https://www.geeksforgeeks.org/grep-command-in-unixlinux/>

SAS reference: <https://blogs.sas.com/content/sgf/2017/12/19/running-sas-programs-in-batch-under-unix-linux/>

Also, I am grateful to Kent Nassen (colleague in Syneos Health) who mentored me in every aspect of UNIX shell scripting as well as SAS programming.

CONTACT INFORMATION

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

Satish Honrao

Syneos Health,

Pune, Maharashtra, India.

Email: satish.honrao@syneoshealth.com

Work Phone: +91- 20485 29 686

Brand and product names are trademarks of their respective companies.