

SmartLog – A UNIX based Utility

```
def _operation == "MIRROR 1":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR 2":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
#mirror_ob.select = 0
#bpy.context.selected_objects[0]
#bpy.data.objects[mirror_ob.name].select = 0
```

Satish Honrao



Introduction

- Satish is currently working as a Manager in Syneos Health situated at Pune, India.
- Satish has 14 years of statistical programming experience in various CROs as well as MNCs.
- Fun Facts about Satish:
 - ψ Satish has travelled on the deck of railway engine alone.
 - ψ Satish has never used WhatsApp till date.



Background

- At the time of submission (interim/final/ad hoc), it is important to submit the programs error-free and within shortest possible time.
- It is really cumbersome job to run individual program at a time and then scan each log for every errors. (unwanted messages and warnings too.)
- In order to execute all above tasks in one go, this UNIX-based utility SmartLog can be invoked so that time and efforts taken will be minimized.

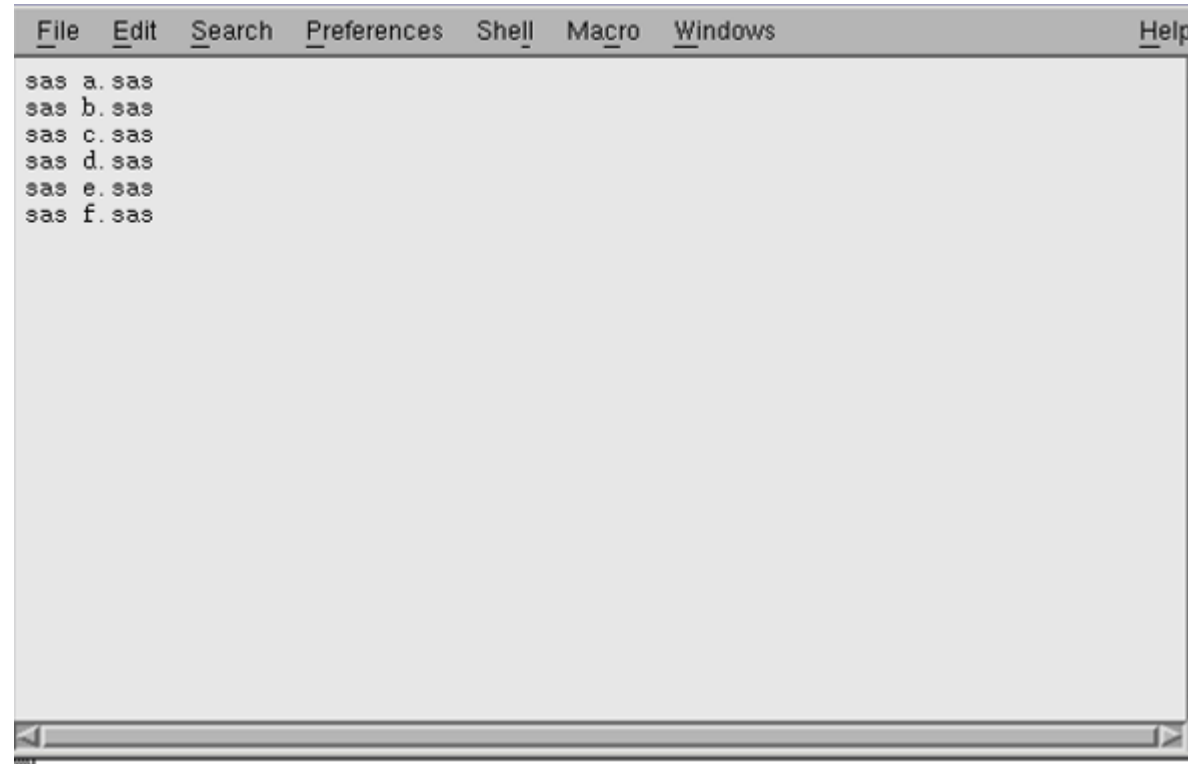


Running all SAS codes

- If the programmer wants to run all SAS codes in one go, one can do that using a simple driver file.
- This driver file will contains all SAS codes present in that folder. (prior to this, make sure that all SAS programs are in working condition.)
- Of course, there are other available ways to run multiple programs in UNIX. But, this is the simplest one to create and execute.



Running all SAS codes (Screenshot)



Running all SAS codes (Executable file)

- As soon as the driver file is created in UNIX, we have to change the access permission of this file as 755.
- This file can be run in either of two ways at the command prompt:
 - (i) *sh drivename*
 - (ii) *./filename*
- Once this file executed, all declared SAS programs will be run one after another in sequential manner and .LOG and .LST will be created for every individual SAS program.



Finding errors, warnings and notes

- In this step, we have to list down all the errors, warnings and notes (unwanted) which need to be highlighted from .LOG files.
- Using “*grep*” command in scripting shell, we can check SAS logs to search for a string of characters in a specific file(s).
- A simple “*grep*” command can be described as: *grep error progname.log*
- This command will search the occurrence of the word “ERROR” in the log and it will print the matching lines to standard output.



Finding errors, warnings and notes (continued..)

- When we are dealing with multiple .LOG files at a time, it is better to use '*grep*' command in a script with different combinations.
- Different messages such as "*Uninitialized*", "*Repeats of*", "*Overwritten*" "*Warning*" etc are combined in the script to scan the log.
- This '*grep*' command is used in *for* loop to scan each and every log present in the folder.



Finding errors, warnings and notes (continued..)

Screenshot for script:

```
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
```



Finding errors, warnings and notes (continued..)

- The script file made executable by modifying access permission mode to “755”.
- This script file can be run on the command prompt in following ways:
 - (i) `scriptname progname.log` (For Single log)
 - (ii) `scriptname *.log` (For Multiple log)
- The result will be immediately displayed at the command prompt.



Finding errors, warnings and notes (continued..)

- The result will be immediately displayed at the command prompt:

```
/temp]$ 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.
/temp]$
```



Collating issues in a single file

- If the error and warning messages are plenty, it will be difficult to accommodate all in a single readable page.
- Hence, the output of the script (i.e. error and other messages) can be re-directed to other temporary file.
- This temporary file will be the dump of all unwanted messages/notes thrown by the script and it can be accessed through command prompt as standalone file.

*scriptname *.log > temp.txt*



Conclusion

- To provide a method of job automation of running multiple SAS programs simultaneously, this utility is useful.
- Main objective of this utility is to keep a track of all errors, warnings and notes of all programs without accessing each and every log files.
- This script is very convenient to use and implement. Hence, it is useful in reduction of time and efforts taken.



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/>



Thank You!



Your comments and questions are valued and encouraged.

Contact the author at:

Satish Honrao

Syneos Health

Building No. 1, CommerZone IT Park, Yerawada, Pune, Maharashtra.

India - 411006.

Email: satish.honrao@syneoshealth.com

Web: <https://www.syneoshealth.com/>

