

## **Log the SAS Log Analyzer for Audit**

Amarnath Vijayarangan, Emmes Service Pvt Ltd

### **ABSTRACT**

The importance of validating a SAS® program through the generated log file is inevitable. A successful execution would require an ERROR, WARNING and any other system message free log. Though, the severity of NOTE or WARNING might not be very high, but there are chances for multiple NOTES or WARNINGS together in a program that can cause severe problems or incorrect results equal to an error message. A programmer must review the log to ensure that every single line of a written program is running successfully without displaying any messages defined by SAS that are potential errors. This means that one needs to carefully review all the logs that have been generated when a program is executed, but a manual review of 1000 + logs becomes taxing and time consuming. For SAS programming audit, it is not only important to analyze the log but also documenting and reporting what is being searched in the log. This paper introduces an approach to log the SAS Log Analyzer.

### **INTRODUCTION**

The SAS programs in various domains generate very large and N number of log files when executed. For instance, Clinical Research domain demands 100 + SAS programs to be either executed in a batch mode or interactive mode for a final delivery, which is validated manually. This means a programmer needs to review 1000 + lines of code in the multiple logs manually where certain seemingly unimportant messages might be overlooked and a manual review is time consuming process.

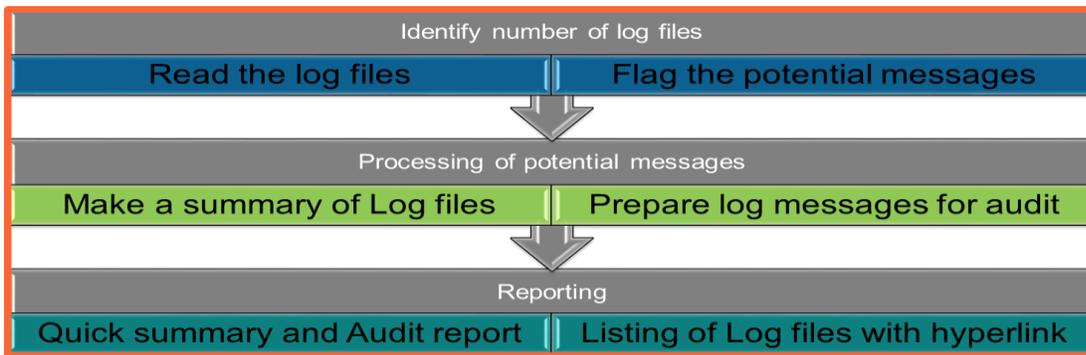
The proposed approach will help in a 360-degree review of the generated SAS Logs with the help of quick summary, potential Log messages being searched; a program is developed to ensure that no system generated message is overlooked. The log analyzer scans each single line of the entire log file in the directory for any system generated messages and provides an overall summary on various system generated messages by SAS logs.

### **SAS PROGRAM FLOW**

SAS macro program is written that performs the following

1. Quick summary of
  - a. # of log files analyzed
  - b. # of log files with issues
  - c. # of log files free from issues.
2. Reporting of the various potential Log messages being searched for Audit
3. Listing of Log files with hyper-link and status of various issue types

For the audit purpose, the list of potential messages has been identified that are recommended by SAS. For each message, short forms are created and detailed text is created. Both the information is reported which helps for anyone to understand what kind of messages are being analyzed. The process flow is explained in the Display 1 as follows.



**Display 1. Process Flow**

## SAS LOG ANALYZER REPORTING

Quick summary of the process. This helps to ensure expected number of files are analysed.

**Log File Summary**  
 Analyzed: 2  
 Issue Free: 0  
 With Issues: 2  
 Report Date: 13MAR2020

Status
Some Log issues found

Reporting of issues for audit/review

### List of Potential Searched Log Messages

Short Form Reported	Detailed Log Text
ERROR	ERROR
MISSING	MISSING VALUES WERE GENERATED
WARNING	WARNING
UNINITIALIZED	UNINITIALIZED
CHAR2NUM	CHARACTER VALUES HAVE BEEN CONVERTED TO NUMERIC VALUES
NUM2CHAR	NUMERIC VALUES HAVE BEEN CONVERTED TO CHARACTER VALUES
INVALIDDATA	INVALID DATA
REPEATS	MERGE STATEMENT HAS MORE THAN ONE DATA SET WITH REPEATS OF BY VALUES
DIVISIONBY0	DIVISION BY ZERO DETECTED
EXTRANEIOUS	EXTRANEIOUS INFORMATION
MATHSOPER	MATHEMATICAL OPERATIONS COULD NOT BE PERFORMED
WDFORMAT	W.D FORMAT
INTERACTIVITY	INTERACTIVITY DISABLED

### List of Log Files with Potential Issues

\*Issue Type is Blank when No Issues Found

Log File Name	Issue Type*
<a href="#">test.log</a>	WARNING
<a href="#">test1.log</a>	ERROR
	WARNING

Listing of logs with hyperlinks for easy access

**Display 2. Report Format**

## CONCLUSION

The proposed approach provides an overall summary, listing of potential messages being analyzed and identifies various system generated messages in SAS logs which will allow the programmer to quicken the log review process ensuring not only accuracy but also a significant reduction in the time taken to review the log files.

## CONTACT INFORMATION

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

Amarnath Vijayarangan  
Emmes Service Pvt Ltd, Bangalore, India  
[avijayarangan@emmes.com](mailto:avijayarangan@emmes.com)  
amarnath7@gmail.com

## APPENDIX

```
/******;  
*Project      : General  
  Program Name : saslogcheck.sas  
  Purpose      : Read SAS Log files and generate a report with Log status for  
each file  
  Author       : Amarnath  
  Date Created : 13MAR2020  
  Software     : SAS 9.4  
  Review Date  :  
  Reviewer Name:  
  Modifications:  
*****;  
*****/  
  
/*  
Keep these two parameters with value as . (dot) to run the program in batch  
from the logs directory, Current folder is used as PATH  
*/  
/*%let logpath =. ;*/  
/*%let outpath=.;*/  
  
/* you can specify path for your logs and output if you want to run from  
other locations:*/  
/*%let logpath=A:\projects;*/  
/*%let outpath=A:\projects;*/  
  
filename pth "&logpath";  
  
data LogList;  
  length PgmName $ 32000;  
  dval=dopen('pth');  
  if dval> 0 then do;  
    count=dnum(dval);  
    do i= 1 to count;  
      PgmName=dread(dval,i);
```

```

        if upcase(scan(PgmName,-1, '.'))='LOG' and upcase(PgmName) not in
: ("SASLOGCHECK" ) then output;
        end;
    end;
keep PgmName;
run;
filename pth clear;
%let Logcnt=0;
data _null_;
    set LogList;
    call symputx("Logcnt",_n_);
run;
proc sort data=LogList;
    by PgmName;
run;

data LogAnalyzer;
    length IssueType $ 20;
    set LogList;
    LogFile=cats("&logpath","\",PgmName);
    infile LogFile filevar=LogFile length=len end=done truncover;
    do while(not done);
        LogFile=LogFile;
        FrequencyCount=1;
        input SystemMessage $varying2000. len ;
        IssueType='';
        SystemMessage=upcase(SystemMessage);
        if scan(SystemMessage,1,':')='ERROR' then IssueType='ERROR';
        if scan(SystemMessage,1,':')='WARNING' then IssueType='WARNING';
        if scan(SystemMessage,1,':')='INFO' then IssueType='INFO';
        if index(SystemMessage,'UNINITIALIZED') then IssueType='UNINITIALIZED';
        if index(SystemMessage,'MISSING VALUES WERE') then IssueType='MISSING';
        if index(SystemMessage,'CHARACTER VALUES HAVE BEEN') then
IssueType='CHAR2NUM';
        if index(SystemMessage,'NUMERIC VALUES HAVE BEEN') then
IssueType='NUM2CHAR';
        if index(SystemMessage,'INVALID DATA FOR') then IssueType='INVALIDDATA';
        if index(SystemMessage,'MERGE STATEMENT HAS MORE THAN') then
IssueType='MERGE';
        if index(SystemMessage,'NOTE: DIVISION BY ZERO DETECTED') then
IssueType='DIVISIONBY0';
        if index(SystemMessage,'EXTRANEIOUS INFORMATION') then
IssueType='EXTRANEIOUS';
        if index(SystemMessage,'W.D FORMAT') then IssueType='WDFORMAT';
        if index(SystemMessage,'REPEATS OF BY VALUES') then IssueType='REPEATS';
        if index(SystemMessage,'MATHEMATICAL OPERATIONS COULD NOT') then
IssueType='MATHSOPER';
        if index(SystemMessage,'INTERACTIVITY DISABLED WITH') then
IssueType='INTERACTIVITY';
        if IssueType^='' then output;
    end;
    keep IssueType SystemMessage PgmName ;
run;

data LogAnalyzer;
    set LogAnalyzer;
    if upcase(PgmName)="SASLOGCHECK" then delete;
run;

```

```

proc sort data=LogAnalyzer nodupkey;
    by PgmName IssueType;
run;
data LogAnalyzer_rpt;
    merge LogList(in=a) LogAnalyzer(in=b);
    by PgmName;
    if a;
run;
data LogAnalyzer_rpt Ulist;
    set LogAnalyzer_rpt;
    by PgmName;
    PgmName=strip("<a href='"||"&logpath"||"\"||strip(PgmName)||strip(''
target='_blank'>")||strip(PgmName)||strip("</a>");
    if not first.PgmName then PgmName='';
    Issue="Yes";
    output LogAnalyzer_rpt;
    if first.PgmName then output Ulist;
run;

proc sql noprint;
    select count(*) into: ncnt from LogAnalyzer;
    select count((PgmName)) into: ncnt_issue from Ulist where IssueType^=''
and PgmName^='';
    select count((PgmName)) into: ncnt_clear from Ulist where IssueType=''
and PgmName^='';
quit;
%put &ncnt;
title;
footnote;
data keys;
length keywords type $ 1000;
infile cards dsd dlm="*";
input keywords type;
cards;
ERROR*ERROR
MISSING*MISSING VALUES WERE GENERATED
INFO*INFO NOTE FOR VARIABLE OVERWRITTEN AND DEFAULTED LENGTH
WARNING*WARNING
UNINITIALIZED*UNINITIALIZED
CHAR2NUM*CHARACTER VALUES HAVE BEEN CONVERTED TO NUMERIC VALUES
NUM2CHAR*NUMERIC VALUES HAVE BEEN CONVERTED TO CHARACTER VALUES
INVALIDDATA*INVALID DATA
REPEATS*MERGE STATEMENT HAS MORE THAN ONE DATA SET WITH REPEATS OF BY VALUES
DIVISIONBY0*DIVISION BY ZERO DETECTED
EXTRANEIOUS*EXTRANEIOUS INFORMATION
MATHSOPER*MATHEMATICAL OPERATIONS COULD NOT BE PERFORMED
WDFORMAT*W.D FORMAT
INTERACTIVITY*INTERACTIVITY DISABLED
MERGE*MERGE STATEMENT HAS MORE THAN
;
run;

data nodata;
    Status="No Log issues found";
run;
data issuedata;
    Status="Some Log issues found";
run;

```

```

data _null_;
  call symputx("today",cats(put(today()),date9.));
run;
%put &today;
proc template;
  define style Styles.saswebm;
    parent = Styles.sasweb;
    style Table /
    rules = ALL
    frame = BOX
    cellpadding = 7
    cellspacing = 1
    background = cxf0f0f0
    foreground = cx002288
    font_style = Roman
    font_weight = Medium
    font_size = 3
    font_face = "Arial, Helvetica, sans-serif";
  end;
run;
ods html close;
%macro rpt;
ods listing close;
ods html path="%outpath" file="saslogcheck.html" style=Styles.saswebm;
title3 "Log File Summary";
title4 h=9pt "Analyzed: &Logcnt";
title5 h=9pt "Issue Free: &ncnt_clear";
title6 h=9pt "With Issues: &ncnt_issue";
title7 h=9pt "Report Date: &today";
%if &ncnt=0 %then %do;
  proc print data=nodata noobs style(header)={just=center};
  run;
  title;
%end;
%if &ncnt^=0 %then %do;
  proc print data=issuedata noobs style(header)={just=center};
  run;
  title;
%end;
title6 "List of Potential Searched Log Messages";
proc print data=keys label noobs style(header)={just=center};
  label keywords="Short Form Reported" type ="Detailed Log Text";
run;
title;
filename body "%outpath\saslogcheck.html" mod;
title3 "List of Log Files with Potential Issues";
title4 h=7pt "*Issue Type is Blank when No Issues Found";
proc print data=LogAnalyzer_rpt label noobs style(header)={just=center};
  var PgmName IssueType ;
  label PgmName='Log File Name' IssueType="Issue Type*";
run;
ods _all_ close ;
ods listing
%mend rpt;
%rpt;

```