Paper 128-31

You've Got E-Mail: Automatic Log Checking Via E-mail Notification

Aaron Augustine, Deloitte & Touche LLP
Prasenjit Dutta, Deloitte & Touche Audit Services India Private Limited

ABSTRACT

Excellence and efficiency in the business world are critical for success. Users are challenged to review SAS® logs accurately in a time sensitive environment. This is often a daunting task when running numerous or lengthy SAS programs. The Log Checking Program is a macro designed to automate the process of reviewing logs by efficiently identifying errors, warnings, and other notes of interest. If problematic messages are found in the review, output reports of the results are sent automatically to the user via e-mail. Without this macro, users are left to perform this task manually and in turn open themselves up to reduced levels of quality and production. This program was designed to run using Base SAS version 9.1.3 in a Windows environment and is intended for intermediate level SAS Users.

INTRODUCTION

Reviewing SAS logs can take up a significant amount of time for a SAS user and programmer. This task becomes cumbersome if one is submitting multiple programs in a batch mode or in succession. In addition, productivity is lost due to multiple iterations of review when a program is resubmitted. Skipping the review of logs is not a good option, because the review of the SAS logs is a key step to support the quality of the SAS analysis. Implementing an automated log checking program can improve both efficiency and quality, by facilitating review of SAS logs and automating tasks that are typically manual and time consuming to perform.

BACKGROUND

Automatically reviewing SAS logs has been a topic of concern for many SAS users. Several solutions for the issue have been presented. For example, during SUGI 27, a paper describing a utility program for checking SAS logs was presented (Carey G. Smoak, Chiron Corporation, Emeryville, CA, SUGI-27, PAPER-96, PDF VERSION). Another example is the Savian Log Analyzer application (SAVIAN, 2004). Although both of these resources are helpful, what makes the Log Checking program more versatile is its ability to automatically accept a SAS log from a prior program as input and provide notification to the user if problems are found in the log. There is no requirement for the user to manually review the SAS log or log check output, although the user does have the option to do an additional review of the log for logic errors that may not be caught with an automated program.

A LOOK AT AUTOMATIC LOG CHECKING VIA E-MAIL NOTIFICATION

OVERVIEW: ADDRESSING THE BUSINESS PROBLEM STATEMENT

Automatic Log Checking via email notification addresses several business needs. The macro can be submitted for as many logs as one needs to review. Automating the log checking process facilitates consistency of the review process and reduces the chance that issues will be missed because the log was manually reviewed. Automating log checking will also boost productivity and facilitate a high quality SAS submission. In the event that an error is detected, notification is automatically provided for follow up.

DETAIL OF THE CODE:

The Log Checking program is set up as a macro with two parameters:

- Name of log to review
- Email ID where the results should be sent.

In order for this macro to run successfully, one should use a PROC PRINTTO statement to write out the log from the SAS programs to an external text file or word document.

The macro begins by reading in the first five characters of each record. In this process, each line in the log is treated as an individual record. By focusing on these characters the macro is able to process the log more efficiently than if it read in the entire record. As the records are read, the macro is specifically looking for the text string 'ERROR',

'WARNI' or 'NOTE:' When either an 'ERROR' or 'WARNI' message is found the code will go on to read the entire record to pull out the text message associated with the error or warning message.

If 'NOTE:' is found the macro will subsequently read and search the entire record specifically for: 'Uninitialized', 'Lost Card', 'New Line', 'Truncated', 'Repeat By Values', and 'Invalid Data' text strings. As with 'ERROR' and 'WARNI', if these are found, the full text message associated with the record is read in. This is accomplished by using a series of trailing '@' to look for key words and '/' to read the next record. Once the log is processed, if any of these problem messages are detected, an email is sent to the given email ID. In the event nothing is found, no action is taken.

The last step in the process defines a global macro variable called LOGCHECK. If a problem message is detected this macro is defined as an asterisk, '*', otherwise it is defined as a blank, ' '. This variable can be extremely helpful when sequentially submitting programs.

For example, a user might want to submit two programs in succession but may only want to submit the second program if no errors were found in the first. To do so, a user could set up the code as follows:

```
%INC "C:\Program1.sas"; RUN;

%LOGCHECK(C:\Program1_Log.txt,EMAILID); RUN;

&LOGCHECK.%INC "C:\Program2.sas"; RUN;

&LOGCHECK.%LOGCHECK(C:\Program2_Log.txt,EMAILID); RUN;
```

If an error is found, the LOGCHECK macro variable resolves to '*' and the line of code that submits program 2 will be treated as a SAS comment.

Another way to employ the Log Checking program would be to imbed the macro at the end of a program.

For example the code might look like this:

```
PROC PRINTTO LOG='testlog.txt' NEW;

DATA ONE;

  X=1;

  Y=2;

RUN;

PROC SORT DATA=ONE; BY X; RUN;

PROC PRINTTO; RUN; /*Reset log back to the log window*/

%LOGCHECK(C:\testlog.txt,emailid); /*submit log check code*/

RUN;
```

SAMPLE OUTPUT

When problems are found in a log, the email report provides several pieces of information that can help in determining what went wrong with the program. It provides a summary report giving the number of errors by keyword. In addition, it includes the log it reviewed as an attachment so that one can immediately identify where the problem occurred.

See example below:

```
ERROR CHECK REPORT

TIME: 9:17                          DATE: 08/31/05

BELOW IS A BREAKDOWN OF THE KEYWORDS FOUND:

KEYWORD                              #OF OCCURENCES

---------                           ----------

ERROR                                        3

THE SPECIFIC MESSAGES ARE AS FOLLOWS

IF YOU ARE UNABLE TO RESOLVE THESE MESSAGES,

PLEASE CONTACT XXXXXX.
```

ORDER#	KEYWORD	DETAILED KEYWORD MESSAGE
001	ERROR	ERROR: DDE session not ready.
002	ERROR	ERROR: Physical file does not exist, Data.xls
003	ERROR	ERROR: File not found.

NOTE:

ATTACHMENT1: THE MAIN LOGFILE BEING CHECKED.

ATTACHMENT2: LOG OF THE LOGCHECK PROGRAM

It is important to note that SAS will send the email report based on the default configuration of the SAS system. For example, MS Outlook might be the default application on ones PC. To change the Email configuration, a user would need to modify the SASV9.CFG file. In a Windows server environment, the setup below might be used:

-EMAILSYS SMTP

-EMAILPORT 25

-EMAILHOST company_SMTP_email_server

-EMAILAUTHPROTOCOL= none

-EMAILID= the_default_sender@company.com

CONCLUSION

Creating high quality results and error free programs has long been a priority of SAS users. Manually reviewing program logs for conditions that could be programmatically detected is inefficient. The Automatic Log Checking program is a key resource, which allows SAS users to more efficiently produce high quality results.

REFERENCES

Savian Log Analyzer, http://www.savian.net/utilities.aspx, Savian, 4/15/2004

Carey G. Smoak, 2002, A Utility Program for Checking SAS Log Files, Proceedings of the Twenty Seventh Annual SAS Users Group International Conference, Chiron Corporation, Emeryville, CA

CONTACT INFORMATION

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

Aaron Augustine
Senior Consultant
Deloitte & Touche LLP
Tel: +1 312 486 3657
aaugustine@deloitte.com
www.deloitte.com

Prasenjit Dutta
Associate Analyst
Deloitte & Touche Audit Services India Private Limited
Tel: +1 615 738 5901
pdutta@deloitte.com
www.deloitte.com

DISCLAIMERS

This publication contains general information only. Deloitte & Touche LLP is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte & Touche LLP, its affiliates, and related entities shall not be responsible for any loss sustained by any person who relies on this publication.

Deloitte refers to one or more of Deloitte Touche Tohmatsu, a Swiss Verein, its member firms and their respective subsidiaries and affiliates. As a Swiss Verein (association), neither Deloitte Touche Tohmatsu nor any of its member firms has any liability for each other's acts or omissions. Each of the member firms is a separate and independent legal entity operating under the names “Deloitte”, “Deloitte & Touche”, “Deloitte Touche Tohmatsu” or other related names. Services are provided by the member firms or their subsidiaries or affiliates and not by the Deloitte Touche Tohmatsu Verein.

Deloitte & Touche USA LLP is the US member firm of Deloitte Touche Tohmatsu. In the US, services are provided by the subsidiaries of Deloitte & Touche USA LLP (Deloitte & Touche LLP, Deloitte Consulting LLP, Deloitte Financial Advisory Services LLP, Deloitte Tax LLP and their subsidiaries), and not by Deloitte & Touche USA LLP.

SOURCE CODE

NOTE: This source code is provided for the purpose of illustrating the points made in this paper. Readers should be encouraged to evaluate and test this source code thoroughly, before deciding to use it in their own SAS programs.

```
%MACRO LOGCHECK(LOGNAME,username);

%GLOBAL LOGCHECK;

DATA ADDRESS(KEEP=X);
LENGTH Y $200.;
Y=SYMGET('LOGNAME');
Z=LENGTH(Y);
X=TRIM(SUBSTR((TRIM(LEFT(Y))),1,Z-4));
RUN;

DATA _NULL_;
SET ADDRESS;
CALL SYMPUT('A',TRIM(X));
RUN;
/* TO STORE LOG AS A TEXT FILE */
PROC PRINTTO LOG="&A._logCheck.txt" NEW;
RUN;

/* Code to include username from logininformation of windows */

DATA USERNAME;
LENGTH X $50.;
X=SYMGET('USERNAME');
USERNAME=TRIM(TRIM(X)||'@company.com');
RUN;

DATA _NULL_;
SET USERNAME;
CALL SYMPUT('MAILID',USERNAME);
RUN;


**********TO CREATE THE DATASET FOR THE ERROR AND OTHER KEYWORD MESSAGES**********;
/*The Code only pulls the first two lines of every log messageif the message contains warning or Error in the
beginning of the line */
/* If line contins Error then this line is held and the full line is read unless */
/* we get '.' Otherwise next line is read. If the desired keyword is found then a flag*/
/* is attached */


DATA LOGCHECK(KEEP= COMPLETE FLAG);
  LENGTH COMPLETE $400. FLAG $14.;
  INFILE "&LOGNAME." TRUNCOVER;
  INPUT @1 LINE $5. @6 LINE1 $200. @;
IF LINE='ERROR' OR LINE='WARNI' THEN DO;
INPUT @1 FULL $200. @;


            /* Code to check error or warning sentences in the log*/
     IF SUBSTR(REVERSE(TRIM(LEFT(FULL))),1,1)='.' THEN DO;
               INPUT;
          COMPLETE=trim(left(FULL));
     END; /****FIRST LINE******/
     ELSE DO;
```

```
            INPUT / NEXT $200. ;
            COMPLETE=TRIM(LEFT(FULL))||' '||TRIM(LEFT(NEXT));
    END;/*****SECOND LINE*****/
    IF LINE='ERROR'  then FLAG="ERROR";
     ELSE IF LINE='WARNI' then FLAG="WARNING";
     OUTPUT;
 END;/****IF A ERROR OR WARNING IS FOUND*****/

 ELSE DO;
 /* Code to check "Note" in the Log*/
          IF LINE='NOTE:' THEN DO;
            INPUT @1 FULL $200. @;
       IF INDEX(FULL,'truncated')>0 or INDEX(FULL,'uninitialized')>0
       OR INDEX(FULL,'lost card')>0  OR INDEX(FULL,'new line')>0
                  OR INDEX(FULL,'repeats of BY values')>0  OR INDEX(FULL,'Invalid data')>0 THEN DO;
              IF SUBSTR(REVERSE(TRIM(LEFT(FULL))),1,1)='.'  THEN DO;
                  COMPLETE=TRIM(LEFT(FULL));

              END;/****FIRST LINE******/

              ELSE DO;
                  INPUT / NEXT $200. ;
               COMPLETE=TRIM(LEFT(FULL))||' '||TRIM(LEFT(NEXT));

              END;/*****SECOND LINE*****/
                   IF INDEX(FULL,'truncated')>0 then FLAG="TRUNCATED";
                    ELSE IF INDEX(FULL,'uninitialized')>0 then FLAG="UNINITIALIZED";
                     ELSE IF INDEX(FULL,'lost card')>0 THEN FLAG="LOSTCARD";
                      ELSE IF INDEX(FULL,'new line')>0 THEN FLAG="NEWLINE";
                              ELSE IF INDEX(FULL,'repeats of BY values')>0 THEN FLAG="REPEATBYVALUES";
                                   ELSE IF INDEX(FULL,'Invalid data')>0 THEN FLAG="INVALIDDATA";
                  OUTPUT;
         END;/****IF A KEYWORD IS FOUND*****/

        END;/*****IF NOTE: IS FOUND****/

    ELSE INPUT;/******IF NOTE IS FOUND BUT NO KEYWORD IS PRESENT IN IT******/

  END;/*****END OF NOTE: BLOCK*****/
RUN;

PROC SORT DATA=LOGCHECK out=LOGCHECK1;
 BY FLAG;
RUN;

************TO Add Serial number in error message************;
DATA LOGCHECK3;
 SET LOGCHECK1;
 LENGTH TEXT2 $22 ;
 COUNT1=_N_;
 IF COUNT1<10 THEN COUNT=COMPRESS(LEFT('00')||COUNT1);
 ELSE IF COUNT1<100 THEN COUNT=COMPRESS(LEFT('0')||COUNT1);
 ELSE COUNT=COUNT1;
 TEXT2=TRIM(LEFT(COUNT))|| "      " || TRIM(LEFT(FLAG));
RUN;


************TO CREATE THE DATASET FOR SUMMMARY************;

PROC SUMMARY DATA=LOGCHECK1 NWAY MISSING;
 BY FLAG;
 OUTPUT OUT=test_sum7(DROP=_TYPE_ RENAME=(FLAG=TEXT1 _FREQ_=COMPLETE));
RUN;
```

```
data test_sum7(drop=x);
set test_sum7(rename=(complete=x));
complete=put(x,best4.);
run;

DATA LOGCHECK3(DROP=COMPLETE count1 count RENAME=(TEXT2=TEXT1 TEXT=COMPLETE));
 SET LOGCHECK3;
 TEXT = COMPLETE;
RUN;

**********TO CREATE MASTER DATASETS WHICH WILL GENERATE DIFFERENT DATASETS TO BE WRITTEN
IN THE MAIL**********;
 DATA STRING9(drop= i);
            LENGTH TEXT1 $ 50 COMPLETE $ 100 ;
            COMPLETE='';
            do i=1 to 6;
            if i=1 then TEXT1= "THE SPECIFIC MESSAGES ARE AS FOLLOWS";
            if i=2 then TEXT1="IF YOU ARE UNABLE TO RESOLVE THESE MESSAGES,";
            if i=3 then TEXT1="PLEASE CONTACT EDA HELPLINE(866-4-EDA-HLP).";
            if i=4 then TEXT1="                                          ";
            if i=5 then do;TEXT1="ORDER#     KEYWORD";
            COMPLETE="DETAILED KEYWORD MESSAGE";end;
            if i=6 then do;TEXT1="-------    --------";
            COMPLETE="                        ----------------------------";end;
            output;
            end;
 RUN;

DATA STRING10(drop= i);
 LENGTH TEXT1 $ 50 COMPLETE $ 100 ;
 COMPLETE='';
 do i=1 to 3;
 if i=1 then TEXT1= "BELOW IS A BREAKDOWN OF THE KEYWORDS FOUND:";
 if i=2 then do;TEXT1="KEYWORD";
 COMPLETE="#OF OCCURENCES";end;
 if i=3 then do;TEXT1="---------";
 COMPLETE="-----------------";end;
 output;
 end;
 run;

DATA CLIENT;
 LENGTH TEXT1 $ 50 COMPLETE $ 200 ;
 TEXT1="ERROR CHECK REPORT";
 COMPLETE='';
RUN;

DATA CLIENT1(DROP=SASTIME TIME DATE HHMM SASDAT DATE1);
 LENGTH TEXT1 $ 50 COMPLETE $ 200 ;
 TIME=TIME();
 DATE=DATE();
 SASTIME=INPUT(TIME,15.);
 HHMM=PUT(SASTIME,TIME5.);
 TEXT1=TRIM(LEFT("TIME:"))||" "||TRIM(LEFT(HHMM));
 SASDAT=INPUT(DATE,15.);
 DATE1=PUT(SASDAT,MMDDYY8.);
 COMPLETE=TRIM(LEFT("DATE:"))||" "||TRIM(LEFT(DATE1));
RUN;

DATA STRING2(drop= i);
 LENGTH TEXT1 $ 50 COMPLETE $ 100 ;
 COMPLETE='';
```

```
 do i=1 to 2;
 if i=1 then TEXT1= "NOTE:";
 if i=2 then TEXT1="ATTACHMENT1:THE MAIN LOGFILE BEING CHECKED.";
 output;
 end;
 run;

DATA STRING7;
 LENGTH TEXT1 $ 50 COMPLETE $ 400;
 TEXT3="ATTACHMENT2:";
  TEXT2="LOG OF THE LOGCHECK PROGRAM";
  TEXT1=TRIM(LEFT(TEXT3))||TRIM(LEFT(TEXT2));
 COMPLETE="";
RUN;

DATA STRING5;
 LENGTH TEXT1 $ 50 COMPLETE $ 200;
 TEXT1="KEYWORD";
 COMPLETE="#OF OCCURENCES";
RUN;

DATA BLANK1;
 LENGTH TEXT1 $ 50 COMPLETE $ 200;
 TEXT1=" ";
 COMPLETE=" ";
RUN;

DATA FINAL;
 SET CLIENT CLIENT1 BLANK1 STRING10 TEST_SUM7 BLANK1 STRING9;
RUN;

DATA MESSAGE1;
SET LOGCHECK3;
LENGTH LAST $ 500;
 LAST=(TEXT1)||" "||LEFT(COMPLETE);
 KEEP LAST;
RUN;


DATA QCFINAL1;
 SET FINAL;
 LENGTH LAST $ 500;
 LAST=(TEXT1)||LEFT(COMPLETE);
 KEEP LAST;
RUN;

DATA QCFINAL2;
 SET BLANK1 STRING2 STRING7 BLANK1 BLANK1;
 LENGTH LAST $ 500;
 LAST=(TEXT1)||LEFT(COMPLETE);
 KEEP LAST;
RUN;

DATA QCFINAL;
SET QCFINAL1 MESSAGE1 QCFINAL2;
RUN;
PROC SQL;
 CREATE TABLE TABLE1 AS SELECT COUNT(*) AS CNT FROM LOGCHECK;
RUN;

DATA _NULL_;
 SET TABLE1;
 CALL SYMPUT('CNT',CNT);
```

```
RUN;

DATA TABLE2;
LENGTH YES $3 NO $3;
YES="*";
NO=" ";
RUN;

************TO CREATE THE MACRO VARIABLE ERRORCHECK************;
DATA _NULL_;
SET TABLE2;
%IF &CNT.>0 %THEN %DO;
CALL SYMPUT('LOGCHECK','*');
%END;
%ELSE %DO;
CALL SYMPUT('LOGCHECK',' ');
%END;
RUN;
%put &LOGCHECK.;

************TO Delete not required DATASETS************;
PROC DATASETS LIBRARY=WORK NOLIST;
  DELETE            ADDRESS
                              BLANK1
                              CLIENT1
                              CLIENT
                              FINAL
                              MESSAGE1
                              LOGCHECK
                              LOGCHECK1
                              LOGCHECK3
                              QCFINAL1
                              QCFINAL2
                              STRING10
                              STRING2
                              STRING5
                              STRING7
                              STRING9
                              TABLE1
                              TABLE2
                              TEST_SUM7
                              USERNAME;
QUIT;

************MACRO TO SEND MAIL************;
%MACRO MAIL;
FILENAME OUTMAIL EMAIL "&MAILID." SUBJECT="LOG CHECK REPORT FOR &LOGNAME."
ATTACH=("&LOGNAME." "&A._logcheck.txt");

DATA _NULL_;
 SET QCFINAL;
 FILE OUTMAIL;
 PUT LAST;
RUN;

%MEND MAIL;

************TO RESET THE OUTPUT LOG LOCATION************;
PROC PRINTTO;
RUN;

DATA _NULL_;
 %IF &CNT.>0 %THEN %DO;
```

```
  %MAIL;
 %END;
RUN;

PROC DATASETS LIBRARY=WORK NOLIST;
  DELETE            QCFINAL;
QUIT;


%MEND LOGCHECK;   /*END OF LOGCHECK MACRO*/
```