



Scanning The Job Log for Errors & Notes

Devendra Patel,
Information Services

Scanning The Job Log for Errors & Notes

Scanning The Job Log – A SAS Process:

This SAS process is a two-stage process:

- Create the Job log in a text file and
- Retrieve the Job log created above and Scan for Error & Warning messages plus Notes.

SOURCE CODE From: SUGI 31 Posters

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.

It also uses PROC PRINTTO process.

Scanning The Job Log for Errors & Notes

Create the Job log in a text file:

SAS program Macros “MC_PRINTTO.sas” will create the job log file in a specified file folder.

SAS program Macros “MC_LogCheck.sas” will scan the file.

Keyword for Scan: 1ST 5 Characters of line

‘ERROR’, ‘WARNI’ & ‘NOTE:’

Within ‘Note:’ ‘There were’, ‘Table’, ‘Physical Name’

Result Log

1. Logfile – few lines of log file (RUN_LoadTB_Refresh_07AUG2013.log)

NOTE: Libref LIB was successfully assigned as follows:

Levels: 1
Engine(1): V9

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/CntlTB/System

NOTE: Libref SYSLIBTB was successfully assigned as follows:

Levels: 1
Engine(1): V9

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/CntlTB/System

SYMBOLGEN: Macro variable LKUPTBPATH resolves to /SAS_data/MSB/Stats_Profiles_Production/Data/LkupTB/

NOTE: Libref LKUPTB was successfully assigned as follows:

Levels: 1
Engine(1): V9

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/LkupTB

NOTE: The query requires remerging summary statistics back with the original data.

NOTE: The query requires remerging summary statistics back with the original data.

.....

More log file lines

.....

NOTE: The infile LIB_IN is:

Filename=/SAS_data/MSB/Stats_Profiles_Production/Data/download/msp_natv_rep.txt,
Owner Name=msbmast1,Group Name=msb_stat_prod,
Access Permission=rwxrw----,
Last Modified=Tue Aug 6 08:01:18 2013,
File Size (bytes)=4878126

Result Log

NOTE: Invalid data for BYEAR in line 1 38-44.

RULE: ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0

1 CHAR hsn,band_resc,res_code,post_code,sex,born_yy,born_mm,age. 57

ZONE 676266665767627675666627677566662767266765772667656626660

NUMR 83EC21E4F2533C253F3F45C0F34F3F45C358C2F2EF99C2F2EFDDC175D

HSN=hsn,band_BAND=resc REscode=res_code PostCD=post_cod SEX=sex BYEAR=. BMONTH=born_mm _ERROR_=1 _N_=1

NOTE: 113261 records were read from the infile LIB_IN.

The minimum record length was 41.

The maximum record length was 57.

NOTE: The data set WORK.W_NAT has 113261 observations and 7 variables.

.....

More log file lines

.....

NOTE: The "<>" operator is interpreted as "not equals".

NOTE: A CASE expression has no ELSE clause. Cases not accounted for by the WHEN clauses will result in a missing value for the CASE expression.

NOTE: A CASE expression has no ELSE clause. Cases not accounted for by the WHEN clauses will result in a missing value for the CASE expression.

WARNING: This CREATE TABLE statement recursively references the target table. A consequence of this is a possible data integrity problem.

NOTE: Table LKUPTB.T_DDLOOKUP created, with 8742 rows and 12 columns.

More log file lines

3. Notes.txt

3. Notes.txt – few lines (RUN_LoadTB_Refresh_07AUG2013_notes.txt)

RUN_LoadTB_Refresh

NOTE: Libref LIB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/CntITB/System

NOTE: Libref SYSLIBTB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/CntITB/System

NOTE: Libref LKUPTB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/LkupTB

NOTE: Libref LKUP_LIB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/LkupTB

NOTE: Libref LKUP_LIB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/LkupTB

NOTE: Libref LKUP_LIB has been deassigned.

NOTE: Table LKUPTB.T_FSC created, with 3514 rows and 4 columns.

NOTE: Table LKUPTB.T_FSC created, with 3514 rows and 5 columns.

NOTE: Table LKUPTB.T_FSC created, with 3514 rows and 6 columns.

NOTE: Libref LKUPTB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/LkupTB

NOTE: Libref INFO was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/msp/info

NOTE: Table LKUPTB.T_LOC created, with 0 rows and 8 columns.

NOTE: Table LKUPTB.T_LOC_DOC created, with 0 rows and 2 columns.

3. Notes.txt

NOTE: Libref LIB_IN was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/download

NOTE: Table LKUPTB.T_PHR_DETAIL created, with 8742 rows and 36 columns.

NOTE: Table LKUPTB.T_PHYS_DETAIL created, with 8742 rows and 36 columns.

NOTE: Table LKUPTB.T_PHR_DOCTORS created, with 3841 rows and 35 columns.

NOTE: Table LKUPTB.T_PHYS_DOCTORS created, with 3841 rows and 35 columns.

NOTE: Table LKUPTB.T_DDLOOKUP created, with 8742 rows and 9 columns.

NOTE: Table LKUPTB.T_DDLOOKUP created, with 8742 rows and 12 columns.

NOTE: Libref STSTB was successfully assigned as follows:

Physical Name(1): /SAS_data/MSB/Stats_Profiles_Production/Data/CntITB/System

NOTE: Table LKUPTB.T_RUNLKUP created, with 20 rows and 10 columns.

NOTE: There were 20 observations read from the data set LKUPTB.T_RUNLKUP.

NOTE: The data set LKUPTB.T_RUNLKUP has 20 observations and 11 variables.

NOTE: Table WORK.REPORTCNTLREC1 created, with 20 rows and 14 columns.

NOTE: Table WORK.REPORTCNTLREC2 created, with 5 rows and 14 columns.

NOTE: There were 20 observations read from the data set WORK.REPORTCNTLREC1.

NOTE: There were 5 observations read from the data set WORK.REPORTCNTLREC2.

NOTE: The data set LKUPTB.REPORTCNTLREC has 25 observations and 14 variables.

NOTE: There were 25 observations read from the data set LKUPTB.REPORTCNTLREC.

NOTE: The data set LKUPTB.REPORTCNTLREC has 25 observations and 14 variables.

2. ErrLog.txt file – Few lines

(RUN LoadTB Refresh 07AUG2013 ErrLog.txt)

LAST

ERROR CHECK REPORT

TIME: 8:33

DATE: 08/08/13

BELOW IS A BREAKDOWN OF THE KEYWORDS FOUND:

KEYWORD #OF OCCURENCES

-----	-----
INVALIDDATA	1
WARNING	4

THE SPECIFIC MESSAGES ARE AS FOLLOWS

IF YOU ARE UNABLE TO RESOLVE THESE MESSAGES,

PLEASE CONTACT EDA HELPLINE(866-4-EDA-HLP).

ORDER# KEYWORD DETAILED KEYWORD MESSAGE

-----	-----
001 INVALIDDATA	NOTE: Invalid data for BYEAR in line 1 38-44.
002 WARNING	WARNING: This CREATE TABLE statement recursively references the target table. A consequence of this is a possible data integrity problem.
003 WARNING	WARNING: This CREATE TABLE statement recursively references the target table. A consequence of this is a possible data integrity problem.
004 WARNING	WARNING: This CREATE TABLE statement recursively references the target table. A consequence of this is a possible data integrity problem.
005 WARNING	WARNING: This CREATE TABLE statement recursively references the target table. A consequence of this is a possible data integrity problem.

How to Create Logfile

SAS Program "How2 Create Logfile.sas":

```
/*=====
```

This code used to create the SAS Log to specified folder & file with system date which can be used to check manually or through MC_LogCheck Scan SAS program for any errors/warning and extract "Notes" information to a file

This code should be used with "MC_PrintTo" SAS program.

MACROS MC_PRINTTO.sas & MC_LogCheck.sas should be put into common source folder so that it can be called in through %includ Statements in any Run time job.

This program modify from to different SAS codes, so the macro run file and JobName should be same

```
=====*/
```

How2_Create_Logfile.sas

```
%INCLUDE "/folder path where source code resides/MC_PRINTTO.sas";  
%let logpth = /SAS_data/...../; /* where folder path file to be saved */  
%let run_file=RUN_MyPgm1; /* Jobname of the program */  
%SETLOGTO (&logpth.&RUN_FILE.);
```

```
/* Your Source code goes here */
```

```
  sas code
```

```
-----
```

```
/* sas code ends here */
```

```
/* last line to reset to normal execution for next job */
```

```
%resetPrintTo;
```

- /* PRINT Report to File */
- %**setPrintTo** (/SAS_data/.../My_Report);
- Report program SAS code
- %**resetPrintTo**;

Retrieve the Job log created above and Scan for Error & Warning messages plus Note.

Check Joblog.sas

```
/*-----  
This program checks the all log file of  
It checks for errors & Warnings as well as creates .txt file for all Notes  
Parameters Required are from the Create log file job  
1. RUN_FILE =RUN_MyPgm1 ; * Specific file name for the run  
2. JOBNAME=&RUN_FILE.; * Assign the jobname  
3. JOBLOG = /where folder path for log file saved/ ; * log path  
4. SYSDATE9 a system date when original job was run e.g. 07AUG2013  
5. Userid User's email id * this feature has not been used yet  
6. Macro LOGCHECK where the source code is in  
%INCLUDE "folder path where source code resides /MC_LogCheck.sas";  
-----*/
```

Retrieve the Job log created above and Scan for Error & Warning messages plus Note

Check Joblog.sas

```
Options obs=max nosymbolgen nomlogic nomprint;
%INCLUDE "/folder path where source code resides/MC_LogCheck.sas";
/* logpath & JOBLOG SAME PATH */
%let JOBLOG = /SAS_data/...../;
/* where folder path for log file saved */
/* SYSTEM DATE RUN */
%let jobdate=07AUG2013; /* &sysdate9.*/
%LET JOBNAME=RUN_MyPgm1;
%LOGCHECK (&joblog.&JOBNAME._&jobdate..log,USERNAME);
run;
```

Sample Test Job

Program:

```
Options obs=max nosymbolgen nomlogic nomprint;
```

```
%INCLUDE
```

```
"/SAS_data/MSB/Stats_Profiles_Production/Source/Macro/Common/MCFReports/MC_LogCheck.sas";
```

```
%Let JOBLOG =
```

```
/SAS_data/MSB/Stats_Profiles_Production/Doc/Log/SP_Main/;
```

```
%let jobdate=07AUG2013; /* &sysdate9. */
```

```
%LET JOBNAME=RUN_LoadTB_Refresh;
```

```
%LOGCHECK (&joblog.&JOBNAME._&jobdate..log,dpatel);
```

```
run;
```

Sample Test Job

```
A./* Create log file to specified folder */  
%let RUN_FILE = MC60;
```

```
%INCLUDE "/SAS_data/MSB/..../Source/Macro/Common/MCFReports/MC_PRINTTO.sas";  
%let logpth = /SAS_data/MSB/Stats_Profiles_Production/Doc/Log/MC60RPT/;  
%SETLOGTO (&logpth.&RUN_FILE.);
```

```
SAS program  
%resetPrintTo;
```

```
B./* create report file to specified folder */  
/* Also create log file for the report */
```

```
%let RUN_log = MC60_07C;  
%SETLOGTO (&logpth.&RUN_log.);  
%setPrintTo (/SAS_data/MSB/...../Data/MW/&QTRT._&cfiscal./07C);  
%RPTDSPRPT (MC6026, 1, 1, &QTRN., 0);  
%resetPrintTo;
```

```
%let RUN_log = MC60_09A_ALL;  
%SETLOGTO (&logpth.&RUN_log.);  
%setPrintTo (/SAS_data/MSB/...../Data/MW/&QTRT._&cfiscal./09A_ALL);  
%RPTDSPRPT (MC6032, 1, 1, &QTRN., 0);  
%resetPrintTo;
```