#SASGF

VIRTUAL
SAS® GLOBAL FORUM 2021

AMERICAS | MAY 18 – 20 ASIA PACIFIC | MAY 19 – 20 EMEA | MAY 25 – 26

Log Reviewing Made Easy

Peter Knapp, U.S. Department of Commerce

For 34 years I have honed base SAS and SAS Macros programming skills for the Federal Government, mostly as a civil servant. At the Department of Commerce, I empower hundreds of international trade analysts to program in SAS, through formal classroom training, one-on-one tutorials, and individual programming assistance and support.

You Will Learn

How to create a Log Report Macro

How it works and why it’s different than the log summary

How to tailor the macro for your specific needs

Anything that appears in the Log can be captured and printed in a Log Report

This Log Report:

- Identifies instances of key words that can be indicative of run problems
- Shows the flow of data through the program

Reviewing the Log for Program Run Problems

✓ One of the golden rules when running SAS® programs is to review the Log for problems

⚠ Using the Log Summary, it's easy to find:

- Syntax Errors
- Warnings

Reviewing the Log for Program Run Problems

- Finding other issues explained in the notes can be more challenging
  - Uninitialized variables
  - Missing values
  - Repeats of by values
  - Converted variables
  - Division by zero detected
  - Invalid data values
  - Rejected missing weight variable values

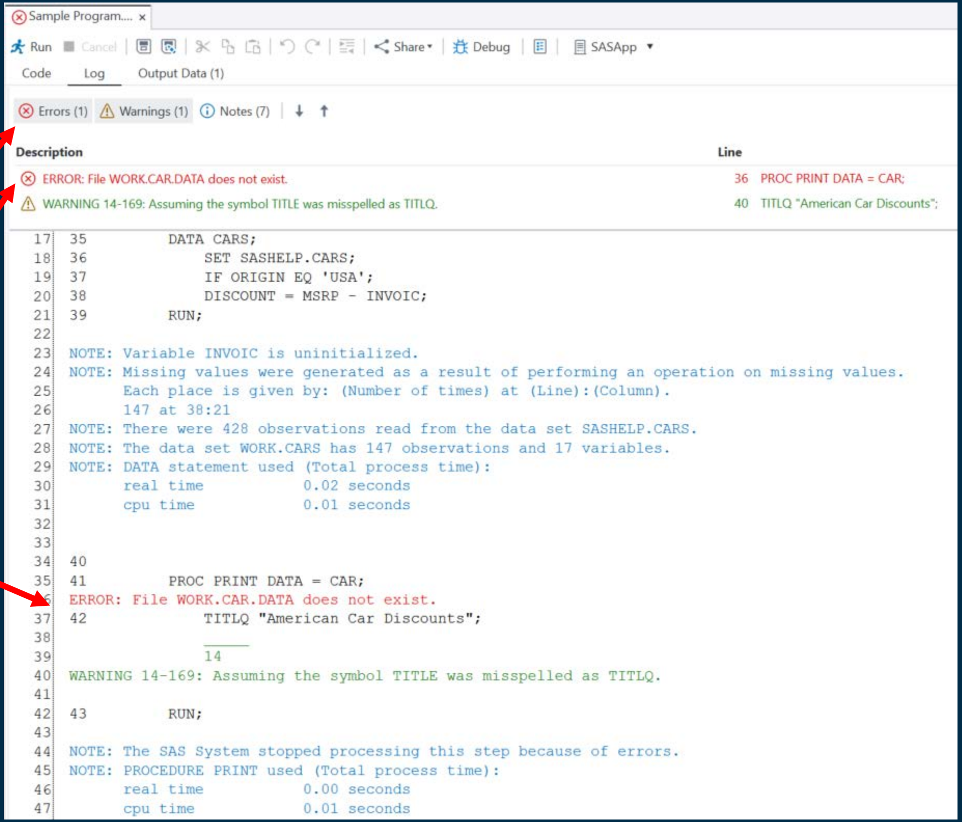
Example Program

- Example program contains:
    - Uninitialized variable
    - Syntax error
    - Warning

```
DATA CARS;
    SET SASHELP.CARS;
    IF ORIGIN EQ 'USA';
    DISCOUNT = MSRP - INVOIC;
RUN;

PROC PRINT DATA = CAR;
    TITLQ "CAR DISCOUNTS";
RUN;
```

Reviewing the Log in SAS Enterprise Guide 8.3

- Syntax Errors
  - Total number displayed in Error tab
  - Listed in Description section
  - ERROR messages in Log

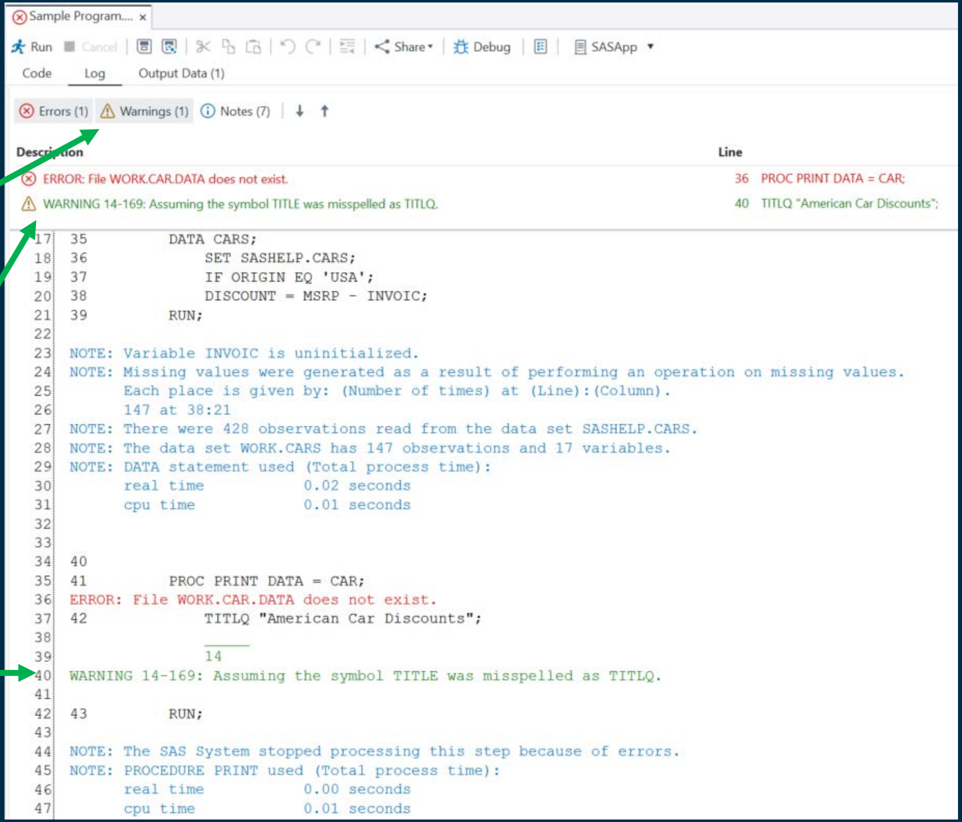


Description	Line	
ERROR: File WORK.CAR.DATA does not exist.	36	PROC PRINT DATA = CAR;
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.	40	TITLQ "American Car Discounts";

```
35          DATA CARS;
36              SET SASHELP.CARS;
37              IF ORIGIN EQ 'USA';
38              DISCOUNT = MSRP - INVOIC;
39          RUN;

NOTE: Variable INVOIC is uninitialized.
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      147 at 38:21
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.CARS has 147 observations and 17 variables.
NOTE: DATA statement used (Total process time):
      real time           0.02 seconds
      cpu time            0.01 seconds


40
41          PROC PRINT DATA = CAR;
ERROR: File WORK.CAR.DATA does not exist.
42              TITLQ "American Car Discounts";
                _____
                14
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.

43          RUN;

NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds
```

Reviewing the Log in SAS Enterprise Guide 8.3

- Warnings
  - Total number displayed in Warning tab
  - Listed in Description section
  - WARNING messages in Log



Description	Line	
ERROR: File WORK.CAR.DATA does not exist.	36	PROC PRINT DATA = CAR;
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.	40	TITLQ "American Car Discounts";

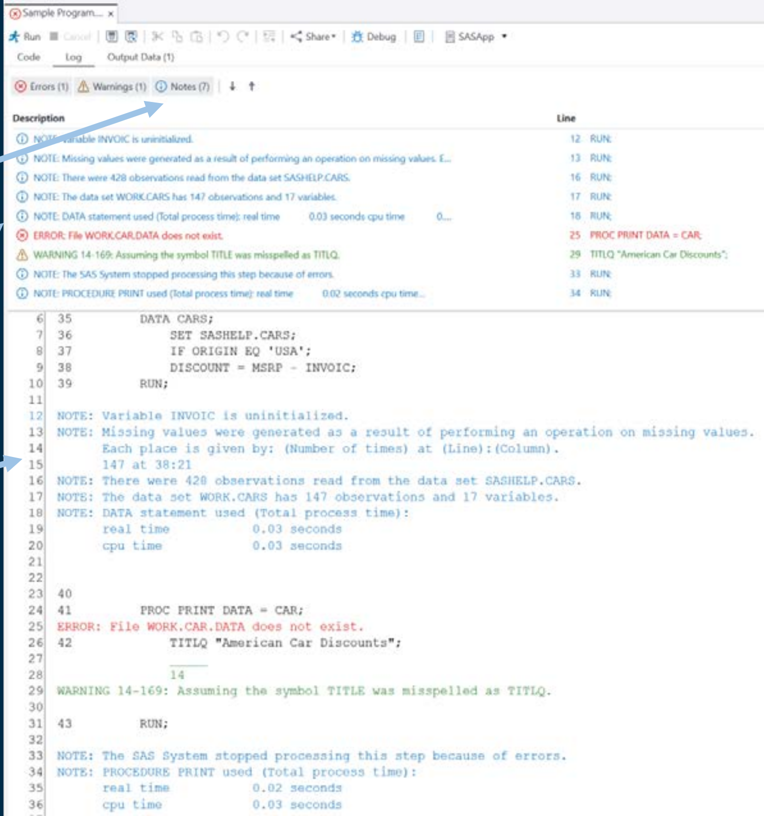
```
17  35         DATA CARS;
18  36             SET SASHELP.CARS;
19  37             IF ORIGIN EQ 'USA';
20  38             DISCOUNT = MSRP - INVOIC;
21  39         RUN;
22
23  NOTE: Variable INVOIC is uninitialized.
24  NOTE: Missing values were generated as a result of performing an operation on missing values.
25        Each place is given by: (Number of times) at (Line):(Column).
26        147 at 38:21
27  NOTE: There were 428 observations read from the data set SASHELP.CARS.
28  NOTE: The data set WORK.CARS has 147 observations and 17 variables.
29  NOTE: DATA statement used (Total process time):
30        real time           0.02 seconds
31        cpu time            0.01 seconds
32
33
34  40
35  41         PROC PRINT DATA = CAR;
36  ERROR: File WORK.CAR.DATA does not exist.
37  42             TITLQ "American Car Discounts";
38                 _____
39                 14
40  WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.
41
42  43         RUN;
43
44  NOTE: The SAS System stopped processing this step because of errors.
45  NOTE: PROCEDURE PRINT used (Total process time):
46        real time           0.00 seconds
47        cpu time            0.01 seconds
```

#SASGF

SAS® GLOBAL FORUM 2021

Reviewing the Log in SAS Enterprise Guide 8.3

- Notes indicative of problems
  - Total number displayed in Notes tab for all Notes
  - Listed in Description section
  - NOTE messages in Log
- Cannot rely on Notes list to know if there are problems with the program run



Description	Line	
NOTE: Variable INVOIC is uninitialized.	12	RUN;
NOTE: Missing values were generated as a result of performing an operation on missing values. E...	13	RUN;
NOTE: There were 428 observations read from the data set SASHELP.CARS.	16	RUN;
NOTE: The data set WORK.CARS has 147 observations and 17 variables.	17	RUN;
NOTE: DATA statement used (Total process time): real time 0.03 seconds cpu time 0...	18	RUN;
ERROR: File WORK.CAR.DATA does not exist.	25	PROC PRINT DATA = CAR;
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.	29	TITLQ "American Car Discounts";
NOTE: The SAS System stopped processing this step because of errors.	33	RUN;
NOTE: PROCEDURE PRINT used (Total process time): real time 0.02 seconds cpu time...	34	RUN;

```
 6  35          DATA CARS;
 7  36              SET SASHELP.CARS;
 8  37              IF ORIGIN EQ 'USA';
 9  38              DISCOUNT = MSRP - INVOIC;
10  39          RUN;
11
12  NOTE: Variable INVOIC is uninitialized.
13  NOTE: Missing values were generated as a result of performing an operation on missing values.
14        Each place is given by: (Number of times) at (Line):(Column).
15        147 at 38:21
16  NOTE: There were 428 observations read from the data set SASHELP.CARS.
17  NOTE: The data set WORK.CARS has 147 observations and 17 variables.
18  NOTE: DATA statement used (Total process time):
19        real time           0.03 seconds
20        cpu time            0.03 seconds
21
22
23  40
24  41          PROC PRINT DATA = CAR;
25  ERROR: File WORK.CAR.DATA does not exist.
26  42              TITLQ "American Car Discounts";
27                  -----
28                  14
29  WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.
30
31  43          RUN;
32
33  NOTE: The SAS System stopped processing this step because of errors.
34  NOTE: PROCEDURE PRINT used (Total process time):
35        real time           0.02 seconds
36        cpu time            0.03 seconds
```

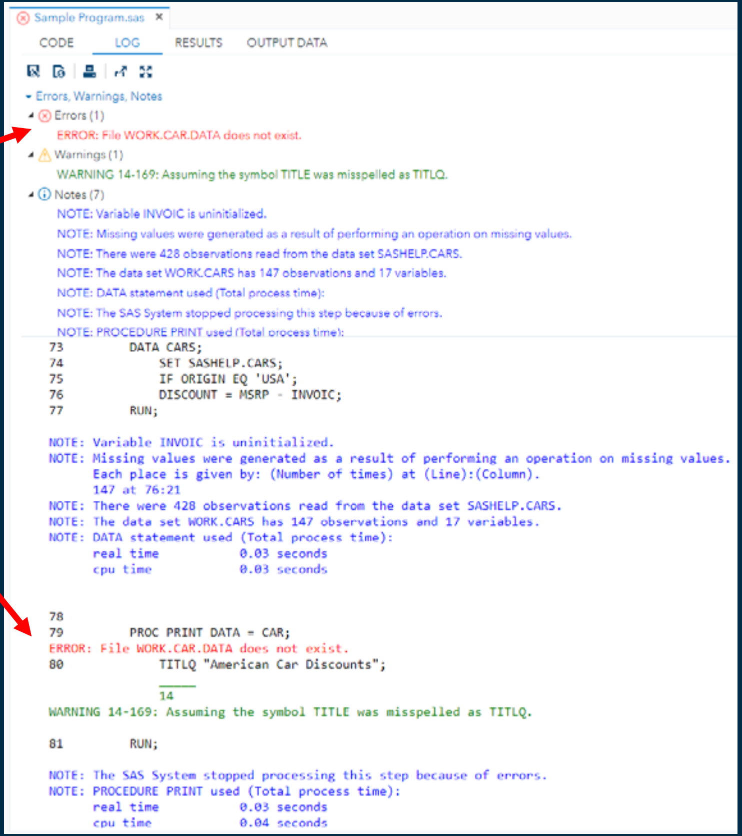
#SASGF

SAS® GLOBAL FORUM 2021

Reviewing the Log in SAS Studio 3.8

- Syntax Errors
  - Total number displayed and explained in the Errors section
  - ERROR messages in Log

- Syntax Errors
- Total number displayed in Error list
- Explained in Description section
- ERROR messages in Log



Errors, Warnings, Notes

Errors (1)

ERROR: File WORK.CAR.DATA does not exist.

Warnings (1)

WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.

Notes (7)

NOTE: Variable INVOIC is uninitialized.

NOTE: Missing values were generated as a result of performing an operation on missing values.

NOTE: There were 428 observations read from the data set SASHELP.CARS.

NOTE: The data set WORK.CARS has 147 observations and 17 variables.

NOTE: DATA statement used (Total process time):

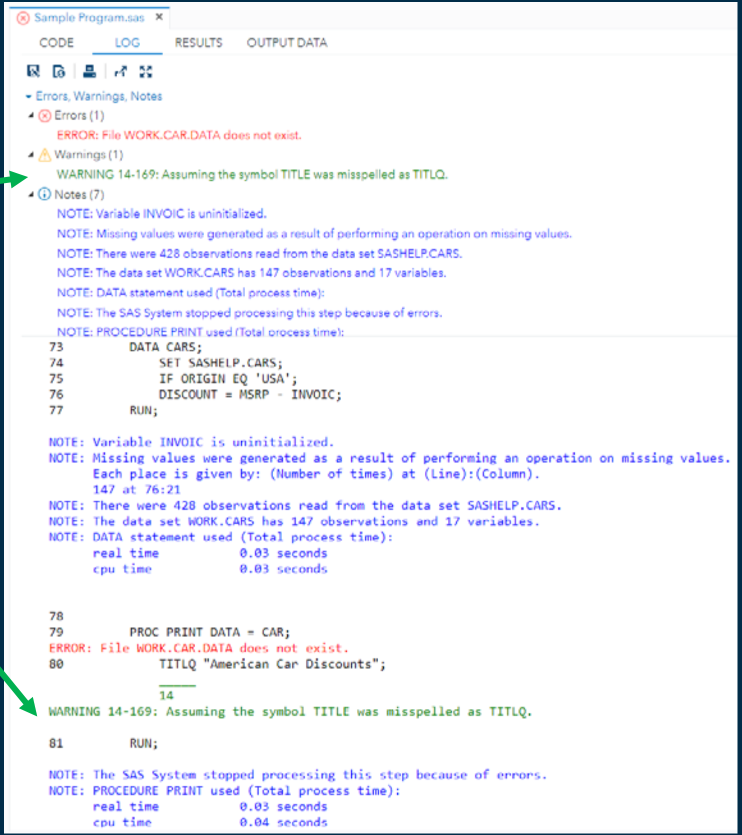
NOTE: The SAS System stopped processing this step because of errors.

NOTE: PROCEDURE PRINT used (Total process time):

```
73         DATA CARS;
74             SET SASHELP.CARS;
75             IF ORIGIN EQ 'USA';
76             DISCOUNT = MSRP - INVOIC;
77         RUN;

NOTE: Variable INVOIC is uninitialized.
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      147 at 76:21
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.CARS has 147 observations and 17 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.03 seconds


78
79         PROC PRINT DATA = CAR;
ERROR: File WORK.CAR.DATA does not exist.
80             TITLQ "American Car Discounts";
               _____
               14
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.

81         RUN;

NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.03 seconds
      cpu time            0.04 seconds
```

Reviewing the Log in SAS Studio 3.8

- Warnings
  - Total number displayed and explained in the Warnings section
  - WARNING messages in Log

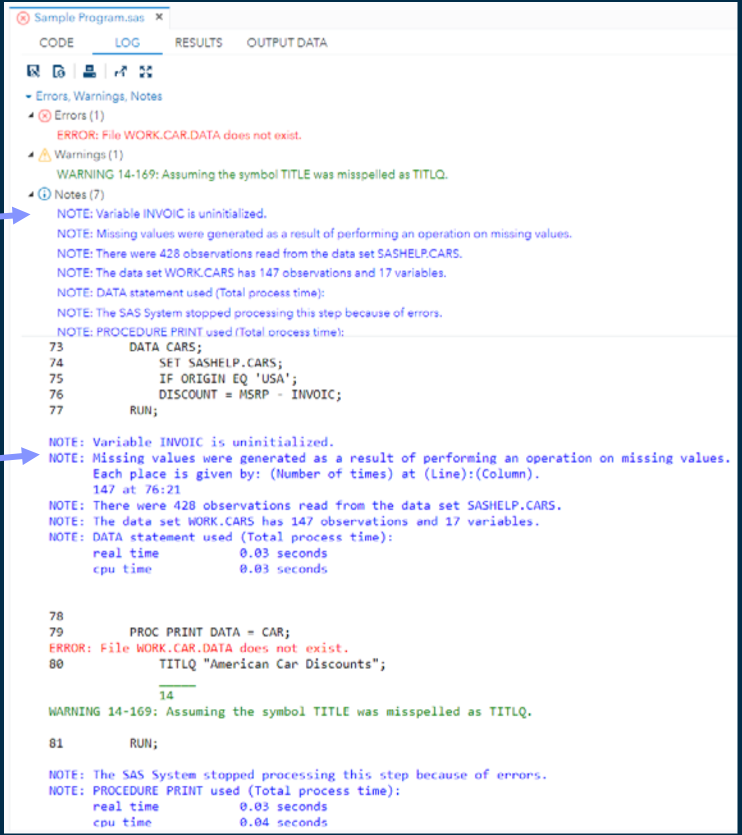


```
ERROR: File WORK.CAR.DATA does not exist.
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.
NOTE: Variable INVOIC is uninitialized.
NOTE: Missing values were generated as a result of performing an operation on missing values.
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.CARS has 147 observations and 17 variables.
NOTE: DATA statement used (Total process time):
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PRINT used (Total process time):
```

```
 73          DATA CARS;
 74              SET SASHELP.CARS;
 75              IF ORIGIN EQ 'USA';
 76              DISCOUNT = MSRP - INVOIC;
 77          RUN;

 NOTE: Variable INVOIC is uninitialized.
 NOTE: Missing values were generated as a result of performing an operation on missing values.
       Each place is given by: (Number of times) at (Line):(Column).
       147 at 76:21
 NOTE: There were 428 observations read from the data set SASHELP.CARS.
 NOTE: The data set WORK.CARS has 147 observations and 17 variables.
 NOTE: DATA statement used (Total process time):
       real time           0.03 seconds
       cpu time            0.03 seconds


 78
 79          PROC PRINT DATA = CAR;
 ERROR: File WORK.CAR.DATA does not exist.
 80              TITLQ "American Car Discounts";
                 _____
                 14
 WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.

 81          RUN;

 NOTE: The SAS System stopped processing this step because of errors.
 NOTE: PROCEDURE PRINT used (Total process time):
       real time           0.03 seconds
       cpu time            0.04 seconds
```

Reviewing the Log in SAS Studio 3.8

- Notes
  - Total number displayed and explained in the Notes section
  - NOTE messages in Log
- Must read all notes in the Notes section to know if there are problems with the program run



```
Errors, Warnings, Notes
  Errors (1)
    ERROR: File WORK.CAR.DATA does not exist.
  Warnings (1)
    WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.
  Notes (7)
    NOTE: Variable INVOIC is uninitialized.
    NOTE: Missing values were generated as a result of performing an operation on missing values.
    NOTE: There were 428 observations read from the data set SASHELP.CARS.
    NOTE: The data set WORK.CARS has 147 observations and 17 variables.
    NOTE: DATA statement used (Total process time):
    NOTE: The SAS System stopped processing this step because of errors.
    NOTE: PROCEDURE PRINT used (Total process time):
```

```
 73          DATA CARS;
 74              SET SASHELP.CARS;
 75              IF ORIGIN EQ 'USA';
 76              DISCOUNT = MSRP - INVOIC;
 77          RUN;

 NOTE: Variable INVOIC is uninitialized.
 NOTE: Missing values were generated as a result of performing an operation on missing values.
       Each place is given by: (Number of times) at (Line):(Column).
       147 at 76:21
 NOTE: There were 428 observations read from the data set SASHELP.CARS.
 NOTE: The data set WORK.CARS has 147 observations and 17 variables.
 NOTE: DATA statement used (Total process time):
       real time           0.03 seconds
       cpu time            0.03 seconds


 78
 79          PROC PRINT DATA = CAR;
 ERROR: File WORK.CAR.DATA does not exist.
 80              TITLQ "American Car Discounts";
                 _____
                 14
 WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.

 81          RUN;

 NOTE: The SAS System stopped processing this step because of errors.
 NOTE: PROCEDURE PRINT used (Total process time):
       real time           0.03 seconds
       cpu time            0.04 seconds
```

The Log Report Is Different Than the Log Summary

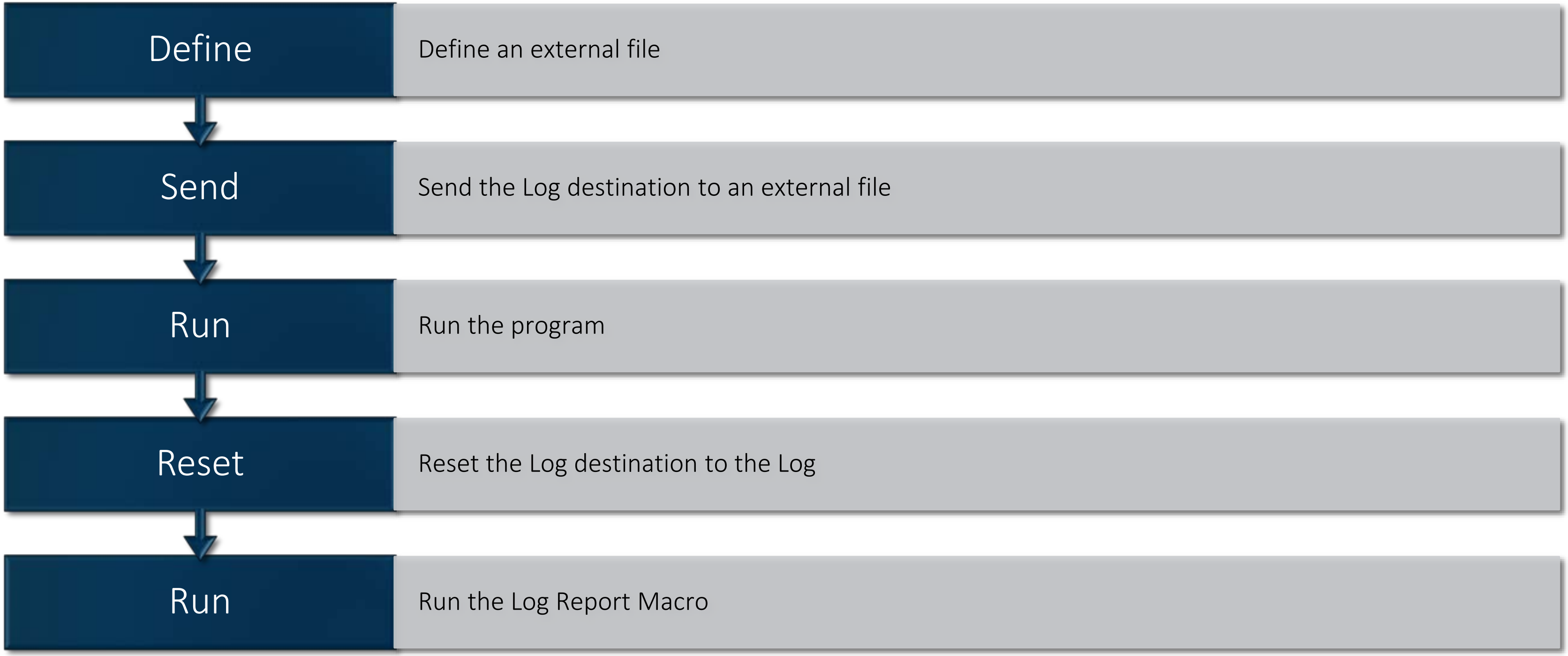
The Log Summary

- Tabulates all notes together
- Need to review all Notes to find notes indicative of problems with the program run

The Log Report

- Identifies specific kinds of Notes indicative of problems with the program run
- No need to review all Notes
- The report can be enhanced to include information about the number of observations flowing through the program

How to Create a Log Report Macro



Define	Define an external file
Send	Send the Log destination to an external file
Run	Run the program
Reset	Reset the Log destination to the Log
Run	Run the Log Report Macro

Define an External File

- Use a FILENAME Statement to:
  - Define the external file path
  - Define the external file name
- Example:

```
FILENAME FILEREF "C:\Sample Program.log";
```

Define an External File

- Alternatively, dynamically define the external file
  - Find path and name of SAS program
  - Replace the program name suffix ‘SAS’ with ‘LOG’
  - Use _SASPROGRAMFILE
    - An automatic macro variable
    - Identifies the path and name of the SAS program

Define an External File

- **%SYSFUNC** macro function
  - Execute SAS functions or user-written functions
- **TRANWRD** macro function
  - Replaces all occurrences of a substring in a character string
- **%UPCASE** macro function
  - Convert values to uppercase
- Example:

```
FILENAME FILEREF "%SYSFUNC(TRANWRD(%UPCASE(&_SASPROGRAMFILE), .SAS, .LOG))";
```

Send the Log Destination to an External File

- Use **PROC PRINTTO**
  - Defines destinations, other than ODS destinations, for SAS procedure output and for the SAS log
- Example:

```
PROC PRINTTO LOG = FILEREF NEW;
RUN;
```

Run the Program

- Example program:

```
DATA CARS;
    SET SASHELP.CARS;
    IF ORIGIN EQ 'USA';
    DISCOUNT = MSRP - INVOIC;
RUN;

PROC PRINT DATA = CAR;
RUN;
```

Reset the Log Destination to the Log

- Use PROC PRINTTO
  - LOG = LOG routes the log to the default log destination
- Example:

```
PROC PRINTTO LOG = LOG;
RUN;
```

Run Log Report Macro

The Log Report Macro does six things:

1. Copies the saved log to the default log destination
2. Reads the saved log
3. Looks for key words
4. Accumulates instances of key words
5. Saves key word totals into macro variables
6. Writes key word totals to the Log Report

Copy the Saved Log to the Default Log Destination

- %MACRO statement
  - Begins the macro definition
- INFILE statement
  - Specifies an external file to read with an INPUT statement
- INPUT statement without arguments
  - Brings an input data record into the input buffer without creating any SAS variables
- PUTLOG statement
  - Writes a message to the SAS log
- _INFILE_ automatic variable
  - Contains the value of the current input record read from a file

```
%MACRO LOG_REPORT;


DATA _NULL_;
    INFILE FILEREF;
    INPUT;
    PUTLOG _INFILE_;
RUN;
```

Read the Saved Log

- END = option
  - Specifies a variable that SAS sets to 1 when the current input data record is the last in the input file
- MISSOVER option
  - Prevents an INPUT statement from reading a new input data record if it does not find values in the current input line for all the variables in the statement
- PAD option
  - Pads the records that are read from an external file with blanks
- Read each line of the Log into the variable LINE

```
DATA _NULL_;
    INFILE FILEREF END = END
           MISSOVER PAD;
    INPUT LINE $250.;
```

Look for Key Words
Accumulate Instances of Key Words

- Look for ‘ERROR:’
  - Use UPPER, COMPRESS, and SUBTR functions to check the first six digits of the variable LINE
  - When found accumulate the variable ERROR
- Look for ‘WARNING:’
  - Use the same three functions to check the first six digits of the variable LINE
  - When found accumulate the variable WARNING
- Look for ‘UNINITIALIZED’
  - Use INDEX and UPCASE functions to look for ‘UNINTIALIZED’
  - When found accumulate the variable UNINIT

```
IF
UPCASE(COMPRESS(SUBSTR(LINE,1,6))) =
"ERROR:" THEN

    ERROR + 1;

ELSE IF
UPCASE(COMPRESS(SUBSTR(LINE,1,8))) =
"WARNING:" THEN

    WARNING + 1;

ELSE DO;

    UNINIT_I =
INDEX(UPCASE(LINE),'UNINITIALIZED');

    IF UNINIT_I THEN

        UNINIT + 1;

END;
```

Save Key Word Totals into Macro Variables

- CALL SYMPUTX Routine
  - Assigns a value to a macro variable and removes both leading and trailing blanks
  - Macro variables created by CALL SYMPUTX are not available until after the DATA Step is run

```
   CALL SYMPUTX('ERROR', ERROR);
   CALL SYMPUTX('WARNING', WARNING);
   CALL SYMPUTX('UNINIT', UNINIT);
RUN;
```

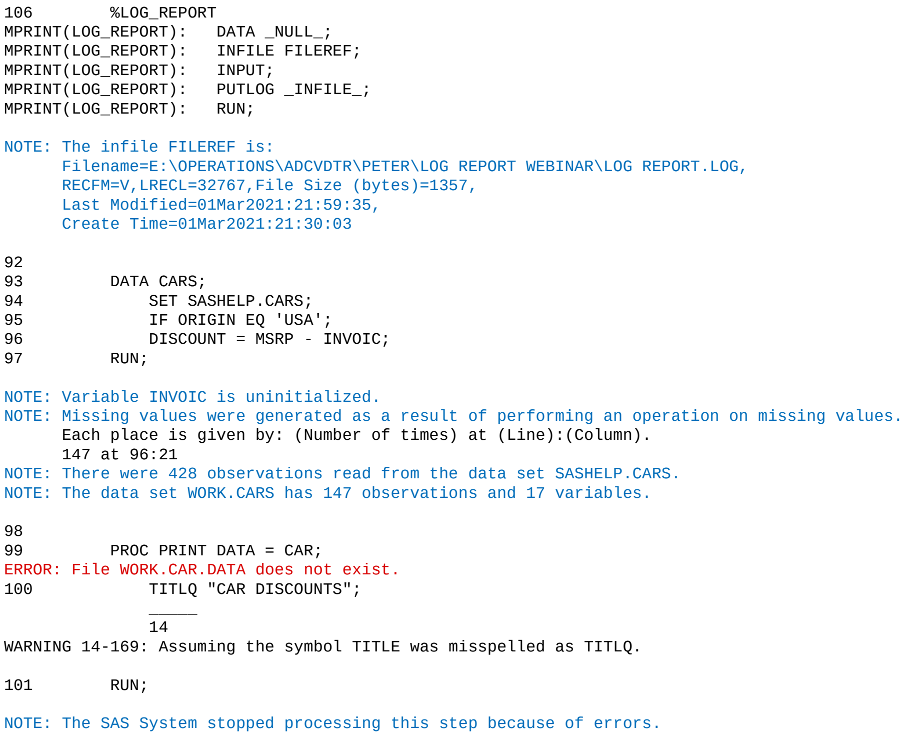
Write Key Word Totals to the Log

- %PUT statement
  - Writes a message to the SAS log
- %MEND statement
  - Ends the macro definition

```
    %PUT **************************************************************;
    %PUT * GENERAL SAS ALERTS: Determine cause of non-zero instances. *;
    %PUT **************************************************************;
    %PUT # OF ERRORS                     = &ERROR;
    %PUT # OF WARNINGS                   = &WARNING;
    %PUT # OF UNINITIALIZED VARIABLES = &UNINIT;
%MEND LOG_REPORT;
```

Results of Log Report Macro Run

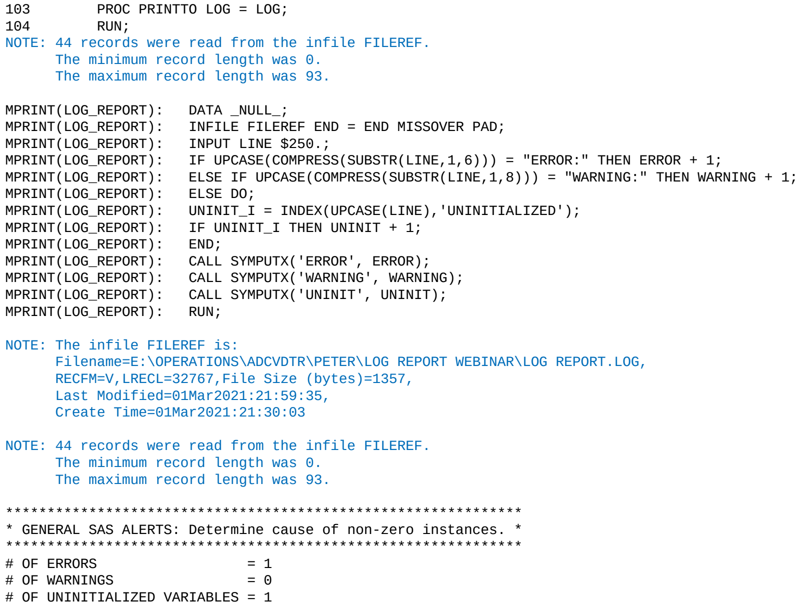
- Copy the saved log to the default log destination
- Read the saved log



```
106        %LOG_REPORT
MPRINT(LOG_REPORT):   DATA _NULL_;
MPRINT(LOG_REPORT):   INFILE FILEREF;
MPRINT(LOG_REPORT):   INPUT;
MPRINT(LOG_REPORT):   PUTLOG _INFILE_;
MPRINT(LOG_REPORT):   RUN;

NOTE: The infile FILEREF is:
      Filename=E:\OPERATIONS\ADCVDTR\PETER\LOG REPORT WEBINAR\LOG REPORT.LOG,
      RECFM=V,LRECL=32767,File Size (bytes)=1357,
      Last Modified=01Mar2021:21:59:35,
      Create Time=01Mar2021:21:30:03

92
93         DATA CARS;
94             SET SASHELP.CARS;
95             IF ORIGIN EQ 'USA';
96             DISCOUNT = MSRP - INVOIC;
97         RUN;

NOTE: Variable INVOIC is uninitialized.
NOTE: Missing values were generated as a result of performing an operation on missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      147 at 96:21
NOTE: There were 428 observations read from the data set SASHELP.CARS.
NOTE: The data set WORK.CARS has 147 observations and 17 variables.

98
99         PROC PRINT DATA = CAR;
ERROR: File WORK.CAR.DATA does not exist.
100            TITLQ "CAR DISCOUNTS";
               _____
               14
WARNING 14-169: Assuming the symbol TITLE was misspelled as TITLQ.

101        RUN;

NOTE: The SAS System stopped processing this step because of errors.
```

Results of LOG Report Macro Run

- Route the log to the default log destination
- Look for key words
- Save key word totals into macro variables
- Write key word totals to the Log Report
    - %PUT statements don’t appear in the LOG
- Log Report



```
103         PROC PRINTTO LOG = LOG;
104         RUN;
NOTE: 44 records were read from the infile FILEREF.
      The minimum record length was 0.
      The maximum record length was 93.

MPRINT(LOG_REPORT):   DATA _NULL_;
MPRINT(LOG_REPORT):   INFILE FILEREF END = END MISSOVER PAD;
MPRINT(LOG_REPORT):   INPUT LINE $250.;
MPRINT(LOG_REPORT):   IF UPCASE(COMPRESS(SUBSTR(LINE,1,6))) = "ERROR:" THEN ERROR + 1;
MPRINT(LOG_REPORT):   ELSE IF UPCASE(COMPRESS(SUBSTR(LINE,1,8))) = "WARNING:" THEN WARNING + 1;
MPRINT(LOG_REPORT):   ELSE DO;
MPRINT(LOG_REPORT):   UNINIT_I = INDEX(UPCASE(LINE),'UNINITIALIZED');
MPRINT(LOG_REPORT):   IF UNINIT_I THEN UNINIT + 1;
MPRINT(LOG_REPORT):   END;
MPRINT(LOG_REPORT):   CALL SYMPUTX('ERROR', ERROR);
MPRINT(LOG_REPORT):   CALL SYMPUTX('WARNING', WARNING);
MPRINT(LOG_REPORT):   CALL SYMPUTX('UNINIT', UNINIT);
MPRINT(LOG_REPORT):   RUN;

NOTE: The infile FILEREF is:
      Filename=E:\OPERATIONS\ADCVDTR\PETER\LOG REPORT WEBINAR\LOG REPORT.LOG,
      RECFM=V,LRECL=32767,File Size (bytes)=1357,
      Last Modified=01Mar2021:21:59:35,
      Create Time=01Mar2021:21:30:03

NOTE: 44 records were read from the infile FILEREF.
      The minimum record length was 0.
      The maximum record length was 93.

**************************************************************
* GENERAL SAS ALERTS: Determine cause of non-zero instances. *
**************************************************************
# OF ERRORS                = 1
# OF WARNINGS              = 0
# OF UNINITIALIZED VARIABLES = 1
```

Enhancing the Log Report Macro For Your Specific Needs

- Anything appearing in the Log can be identified and added to the Log Report Macro
- For example, the Log Report Macro can
  - Identify how many observations are read in from a data set
  - Identify how many observations are kept

Example Program

- Example Program
  - Reads the CARS dataset from the SASHELP library
  - Keeps cars originating in the USA

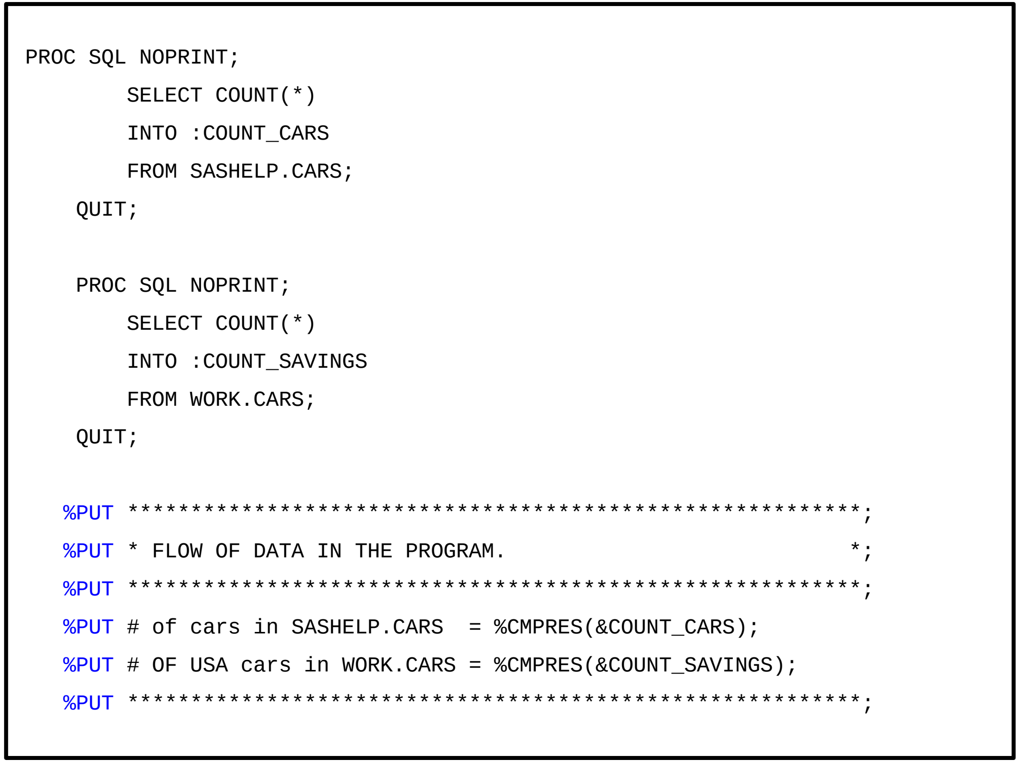
```
    DATA CARS;
          SET SASHELP.CARS;
          IF ORIGIN EQ 'USA’;
    RUN;
```

Added Functionality to The Log Report Macro

- Look for data set observations numbers
- Save data set observations numbers into macro variables
- Write data set observations numbers to Log Report

Identify and Print Data Set Observations

- PROC SQL
  - SELECT total number of observations
  - FROM specified data set
  - Assign totals INTO macro variables
- %PUT statement
  - Writes a message to the SAS log



```
PROC SQL NOPRINT;
        SELECT COUNT(*)
        INTO :COUNT_CARS
        FROM SASHELP.CARS;
    QUIT;

    PROC SQL NOPRINT;
        SELECT COUNT(*)
        INTO :COUNT_SAVINGS
        FROM WORK.CARS;
    QUIT;

   %PUT **********************************************************;
   %PUT * FLOW OF DATA IN THE PROGRAM.                           *;
   %PUT **********************************************************;
   %PUT # of cars in SASHELP.CARS  = %CMPRES(&COUNT_CARS);
   %PUT # OF USA cars in WORK.CARS = %CMPRES(&COUNT_SAVINGS);
   %PUT **********************************************************;
```

Results of Log Report Macro Run

- SELECT total number of observations FROM specified data set and Assign totals INTO macro variable
- Write key word totals to the Log Report
  - %PUT statements don’t appear in the LOG
- Log Report
  - Bottom of LOG

```
MPRINT(LOG_REPORT):   PROC SQL NOPRINT;
MPRINT(LOG_REPORT):   SELECT COUNT(*) INTO :COUNT_CARS FROM SASHELP.CARS;
MPRINT(LOG_REPORT):   QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds


MPRINT(LOG_REPORT):   PROC SQL NOPRINT;
MPRINT(LOG_REPORT):   SELECT COUNT(*) INTO :COUNT_SAVINGS FROM WORK.CARS;
MPRINT(LOG_REPORT):   QUIT;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```

```
**************************************************************
* GENERAL SAS ALERTS: Determine cause of non-zero instances. *
**************************************************************
# OF ERRORS                     = 1
# OF WARNINGS                   = 0
# OF UNINITIALIZED VARIABLES = 1
**************************************************************
* PROGRAM FLOW OF DATA IN THE PROGRAM: Obs before and after. *
**************************************************************
# OF CARS IN SASHELP.CARS   = 428
# OF USA CARS IN WORK.CARS = 147
```

Conclusion

- The Log Summary is a great way to identify ERRORs and WARNINGs but not NOTEs indicative of problems
- The Log Report identifies specific kinds of NOTEs indicative of problems with the program run
- The Log Report can be enhanced to identify anything the appears in the Log Summary

References

- SAS® 9.4 and SAS® Viya® 3.5 Programming Documentation. Available at https://documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=pgmsashome&docsetTarget=home.htm&locale=en
- Knapp, Peter. 2020. "Log Reviewing Made Easy" Proceedings of the Southeast SAS User Group 2020 Conference, Savannah, Georgia. Available at https://sesug.org/proceedings/sesug_2020_final_papers/Know_Your_SAS__Advanced_Techniques/SESUG2020_Paper_112_Final_PDF.pdf

Acknowledgments

I would like to thank my colleague Girish Narayandas who adapted and improved the Log Report Macro I originally wrote to work with SAS Enterprise Guide.

Thank you!

Your comments and questions are valued and encouraged.

Peter Knapp
U.S. Department of Commerce
Peter.Knapp@trade.gov
202-482-1359