

Navigate the SAS® Log – GPS Style

Rebecca Ottesen, California Polytechnic State University, San Luis Obispo, CA

Ethan Miller, SRI International, Menlo Park, CA

ABSTRACT

Sometimes the notes, warnings and errors in the SAS Log can be cryptic at best. Hours of programming and deciphering the log can make a person feel a little down and somewhat nutty. What if there was a way to have the SAS log be informative and amusing at the same time? Having the option to change how the SAS log communicates might actually keep a user from throwing their computer out the window. Our aim is to help thousands of SAS programmers to understand how the messages in the log can be interpreted in an entertaining way.

INTRODUCTION

What does the SAS log tell us and why is it important? We all know that the SAS log is used to provide us important information about the compilation and execution of our programs. We use it for many reasons such as debugging code, verifying the reading of a data set, checking for automatic character or numeric conversions, verifying macro syntax, and identifying ODS objects. The SAS language is very precise and the purpose of the SAS log is to be helpful in identifying programming errors. The log will even color code errors, warnings and notes, and provide messages, which often include suggestions to help remedy the problem.

Unfortunately deciphering these log messages can be challenging even for an experienced programmer. In addition, imagine spending hours, or days, debugging a complex program where even 10 cups of coffee can't make the log any easier to interpret. Programming experience, colleagues, tech support and the internet may be able to help, but they won't be able to make you laugh (although it depends on their sense of humor.) The one thing that might keep you sane is a GPS style SAS log where you are able to pick the communication style of your SAS log messages.

We would like to introduce the SAS log with character representations by the: drill sergeant, strict nun school teacher, don't mess with me judo lady, short and sweet, voice of the heavens, British lady, redneck guy, valley girl, and Yoda.

SYNTAX ERRORS

```
1  DATA new;
2  INPUT a b c
3  DATALINES;
4  1 2 3
   -
   180
ERROR 180-322: Statement is not valid or it is used out of proper order.
```

```
5  4 5 6
6  3 6 9
7  ;
8  RUN;
```

ERROR: No DATALINES or INFILE statement.

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

WARNING: The data set WORK.NEW may be incomplete. When this step was stopped there were 0 observations and 4 variables.

These error messages tell us that SAS could not find any data to read, either by the method of instream data input using DATALINES or with an INFILE statement. This is confusing because we do have a DATALINES statement and the first error message indicates that the first line of raw data is out of order. The root cause of the real error is that there is no semi-colon at the end of the INPUT statement, which causes SAS to not see the DATALINES statement. Let's see how a GPS style log would explain this error message.

Character	Log Message
SAS Log	ERROR: No DATALINES or INFILE statement.
Drill sergeant	ERROR: Where is my data? Don't make me interrogate your DATA step.
Strict nun school teacher	ERROR: Your data set is envious of DATALINES or INFILE. In this case, envy is appropriate.
Don't mess with me judo lady	ERROR: No DATALINES or INFILE? I'll blow my top. Don't make me throw you.
Short and sweet	ERROR: SAS can't see the DATA LINE or INFILE statement.
Voice of the heavens	ERROR: Thy DATA step lackith a DATA LINE or INFILE statement.
British lady	ERROR: No DATALINES or INFILE statement? That wasn't very clever.
Redneck guy	ERROR: Something in the milk ain't clean, your INPUT got no data.
Valley girl	ERROR: No DATALINES or INFILE statement, as if!
Yoda	ERROR: No DATALINES or INFILE statement. Seek advice, you must.

Table 1. GPS log interpretation for a syntax error log message

In the same log above there was also a warning message. This is a warning and not an error because it may or may not produce the result that we intended. This warning tells us that there are no observations in our data set named new, and more than likely, this is not what we wanted for the resulting data set.

Character	Log Message
SAS Log	WARNING: The data set WORK.NEW may be incomplete. When this step was stopped there were 0 observations and 4 variables.
Drill sergeant	WARNING: Your data is unacceptable. It has no variables. Where are my variables?
Strict nun school teacher	WARNING: On the 8th day the lord said let there be variables.
Don't mess with me judo lady	WARNING: Programming requires discipline. It also requires variables, so check your data set or you'll pay.
Short and sweet	WARNING: Your data set has no variables.
Voice of the heavens	WARNING: Thy data set is devoid of variables.
British lady	WARNING: Chop chop check your data set straight away, it appears to be missing all the bloody variables.
Redneck guy	WARNING: You ain't got no variables.
Valley girl	WARNING: Like oh my god, where are your variables?
Yoda	WARNING: Incomplete your data is. Your variables, where are, hmmm?

Table 2. GPS log interpretation for a syntax error log message

MISSING VALUES

Once the error is resolved and the data is finally in a SAS data set the program is ready to carry out more data manipulation. Suppose we add a simple statement to find the total of the numeric variables in this data set. We're feeling pretty good with the log messages below, no red error messages, right? But upon further investigation of the output we notice that the `total` variable has all missing data (Output 1) as is also the case for the variable called `d`. This is the result of there being no variable called `d` in the DATA set named `new`, therefore when SAS uses this uninitialized variable (or even an existing variable with missing values) in a simple addition statement the result will be missing. SAS tells us this in a note because it may or may not be what we wanted.

```
9 DATA vars; SET new;
10 total=a+b+c+d;
11 RUN;
```

NOTE: Variable d 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).
3 at 10:12

NOTE: There were 3 observations read from the data set WORK.NEW.

NOTE: The data set WORK.VARS has 3 observations and 5 variables.

Obs	a	b	c	total	d
1	1	2	3	.	.
2	4	5	6	.	.
3	3	6	9	.	.

Output 1. Results from a DATA step with an uninitialized variable

Character	Log Message
SAS Log	NOTE: Variable d is uninitialized.
Drill sergeant	NOTE: You are referencing an AWOL variable. You need to locate this variable immediately.
Strict nun school teacher	NOTE: Reference the right variable or I'll smart you.
Don't mess with me judo lady	Note: Your variable d may have been flipped out of your data set.
Short and sweet	NOTE: The variable d does not exist.
Voice of the heavens	NOTE: The variable which you reference, d, is in the ether and not the data set.
British lady	NOTE: Dear, if you don't mind me saying the variable d appears to be missing.
Redneck guy	NOTE: Your SAS ain't never heard of that variable.
Valley girl	NOTE: Variable d is not there, gag me with a spoon.
Yoda	NOTE: Uninitialized, variable d is.

Table 3. GPS log interpretation for a missing value log message

Suppose that the missing data was not a column of missing values but instead a single missing value as shown for the variable `a` in Output 2 below. Creating the `total` variable with the sum of the variables `a`, `b` and `c` will still result in a note to the log as shown below. This data may in fact be acceptable, it depends on the setting, but we should always verify missing value notes.

```

12 DATA vars; SET new;
13 total=a+b+c;
14 RUN;

```

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).
1 at 13:8

NOTE: There were 3 observations read from the data set WORK.NEW.

NOTE: The data set WORK.VARS has 3 observations and 4 variables.

Obs	a	b	c	total
1	1	2	3	6
2	.	5	6	.
3	3	6	9	18

Output 2. Results from a DATA step with missing values in a variable

Character	Log Message
SAS Log	NOTE: Missing values were generated as a result of performing an operation on missing values.
Drill sergeant	NOTE: Your new variable is missing for some observations. Your mama is not here to double-check your data, double check it now!
Strict nun school teacher	Note: Type this in Notepad 1000 times - I will not perform math operations on missing values.
Don't mess with me judo lady	Note: Missing values in, missing values out.
Short and sweet	NOTE: Some of your observations have missing values.
Voice of the heavens	NOTE: Thy operation hath resulted in missing data.
British lady	NOTE: Oh fancy that! Your new data set has missing values.
Redneck guy	NOTE: Your operation done made some variables empty.
Valley girl	NOTE: Barf me out! Your new data set, like, totally has missing data.
Yoda	NOTE: Generated as a result of performing an operation on missing values, missing values were. Yes, hmmm.

Table 4. GPS log interpretation for a missing value log message

AUTOMATIC CONVERSIONS

A common issue with SAS DATA step programming is accidentally handling a character variable as numeric or a numeric variable as character. Even experienced programmers are not immune to this mistake because SAS is generally very forgiving and converts the data for us. This will generate an automatic conversion note in the log which is not good form, but there is a simple solution available. We can either correct the programming statement that is causing conversion, as represented in the code below `a=' '`, or we can use an INPUT() or PUT() function to carry out the conversion. This way we can guarantee our results and we are not asking SAS to make a decision for us. In this example the code in line 16 refers to a missing numeric variable as if it were a character. Removing the quotes would from the missing numeric value will solve the automatic conversion problem shown in the log.

```

15 DATA chars; SET new;
16 IF a='.' THEN missing='yes';
17 ELSE missing='no';
18 RUN;

```

NOTE: Character values have been converted to numeric values at the places given by: (Line):(Column).
16:6

NOTE: There were 3 observations read from the data set WORK.NEW.

NOTE: The data set WORK.CHARS has 3 observations and 4 variables.

Character	Log Message
SAS Log	NOTE: Character values have been converted to numeric values at the places given by: (Line):(Column). 16:6
Drill sergeant	NOTE: Be advised that your character variables were converted to numeric at the following coordinates 16:6
Strict nun school teacher	NOTE: For being slothful and making SAS convert your character values to numeric values, say 10 Our Fathers.
Don't mess with me judo lady	NOTE: SAS grappled your character values into submission; they are now numeric.
Short and sweet	NOTE: SAS converted your data to the proper type.
Voice of the heavens	NOTE: That which once was character is now numeric.
British lady	NOTE: SAS has had a bit of a rethink. What we had been led to expect was a character variable is actually numeric.
Redneck guy	NOTE: Your letters now done become numbers.
Valley girl	NOTE: What's the deal? Are your variables like numeric, or like character? Oh my god!
Yoda	NOTE: Been converted to numeric values at the places given by, character values have. Herh herh herh.

Table 5. GPS log interpretation for an automatic conversion log message

CONFLICTING VARIABLE TYPES

When joining data sets with a merge it is important that same named variables, which are often key ids, be the same variable type. If the variables are not considered a key id the solution could be to rename one of the variables. If the variables are a key id then the solution would be to convert one of the variables types before merging by using a PUT() or INPUT() function. In this latter case SAS will issue an error to alert us to the problem as shown in the log below.

```

19 DATA join;
20 MERGE dsn1
21     dsn2;
ERROR: Variable id has been defined as both character and numeric.
22 BY id;
23 RUN;

```

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

WARNING: The data set WORK.JOIN may be incomplete. When this step was stopped there were 0 observations and 5 variables.

WARNING: Data set WORK.JOIN was not replaced because this step was stopped.

Character	Log Message
SAS Log	ERROR: Variable id has been defined as both character and numeric.
Drill sergeant	ERROR: Your variable types are not consistent; fix this or you'll be doing push-ups for days.
Strict nun school teacher	ERROR: There'll be no merging until your variable types match.
Don't mess with me judo lady	ERROR: Fix your variable types or I'll fix you.
Short and sweet	ERROR: To correctly merge the key variable types must be the same.
Voice of the heavens	ERROR: Thy merge requires variables of the same type.
British lady	ERROR: Oh dear, that's dreadful. Your variable id needs to match in both data sets.
Redneck guy	ERROR: Now how you fittin' to merge them variables when they ain't the same type?
Valley girl	ERROR: Like, H-O-W is your merge going to work with different variable types? Oh my god, duh!
Yoda	ERROR: Been defined as both character and numeric, variable id has.

Table 6. GPS log interpretation for conflicting variable types log message

DROP, KEEP AND RENAMING VARIABLES

When using DROP, KEEP and RENAME statements with variables that do not exist, SAS will complete the DATA step, but will issue a warning about any variables that have not been referenced. This message is confusing because we are referencing these variables, but they do not exist in the data set. Typically this message is a result of a typo in the spelling of a variable name as shown in the log below. In this case the resulting data set will only keep the legitimate variables that appeared in the KEEP statement (Output 3).

```

24 DATA vars; SET new;
25 KEEP a d c;
26 RUN;

```

WARNING: The variable d in the DROP, KEEP, or RENAME list has never been referenced.

NOTE: There were 3 observations read from the data set WORK.NEW.

NOTE: The data set WORK.VARS has 3 observations and 3 variables.

Obs	a	c
1	1	3
2	.	6
3	3	9

Output 3. Results from a DATA step with an incorrect KEEP statement

Character	Log Message
SAS Log	WARNING: The variable d in the DROP, KEEP, or RENAME list has never been referenced.
Drill sergeant	WARNING: AWOL variables cannot be referenced in a KEEP, RENAME, or DROP statement; Now, drop and give me 50 NOW!
Strict nun school teacher	WARNING: You will receive a "D" if you do not remove the undefined variable d from your DROP, KEEP, or RENAME statement.
Don't mess with me judo lady	WARNING: d is for dropkick, not for your DROP, KEEP, or RENAME statement. If you want to reference it, then define it.
Short and sweet	WARNING: The variable d in the KEEP statement does not exist.
Voice of the heavens	WARNING: SAS knoweth not of the variable d of which you speaketh.
British lady	WARNING: Oh looks like you made a right mess of that. SAS has no idearr what variable you're speaking of.
Redneck guy	WARNING: SAS don't know what variables you's talking about.
Valley girl	WARNING: I am so sure, SAS totally can't find your variable.
Yoda	WARNING: The variable d in the drop, keep, or never been referenced, rename list has. Yeesssssss.

Table 7. GPS log interpretation for DROP, KEEP and RENAME log messages

ARRAY MESSAGES

Common mistakes when working with arrays are naming the array the same name as a SAS defined function and getting an array subscript out of range error. When we name an array the same as a function it renders the function useless, which is not advisable as shown in the log below.

```

27 DATA array; SET new;
28 ARRAY nmiss(3) a b c;

```

NOTE: The array nmiss has the same name as a SAS-supplied or user-defined function. Parentheses

following this name are treated as array references and not function references.

```

29 DO i=1 TO 4;
30 IF nmiss(i)=. THEN flag=1;
31 END;
32 RUN;

```

ERROR: Array subscript out of range at line 30 column 5.

a=1 b=2 c=3 i=4 flag=. _ERROR_=1 _N_=1

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

NOTE: There were 1 observations read from the data set WORK.NEW.

WARNING: The data set WORK.ARRAY may be incomplete. When this step was stopped there were 0 observations and 5 variables.

WARNING: Data set WORK.ARRAY was not replaced because this step was stopped.

Character	Log Message
SAS Log	NOTE: The array nmiss has the same name as a SAS-supplied or user-defined function.
Drill sergeant	NOTE: Programmer, you are not authorized to use SAS keywords as your own. You are out of line.
Strict nun school teacher	NOTE: Stealing keywords from SAS to use as your own is the eighth deadly sin.
Don't mess with me judo lady	NOTE: If you try to use SAS keywords again, you will be nmissing some teeth.
Short and sweet	NOTE: Your array needs a new name. Please choose wisely.
Voice of the heavens	NOTE: Thy array name creativity is lacking.
British lady	NOTE: Dear dear, I'm not particularly fond of your array name, it is the same as a SAS function name.
Redneck guy	NOTE: Just like catchin' a weasel, you got to be uniquely inventitve with your array names.
Valley girl	NOTE: Your array has the same name as a function, totally gnarly.
Yoda	NOTE: The same name as the SAS-supplied or user-defined function, the array nmiss has.

Table 8. GPS log interpretation for an array log message

An array subscript out of range error tells us that we have referred to an incorrect dimension in the array call. This is typically the result of an incorrectly specified index in a DO loop as shown in the previous log.

Character	Log Message
SAS Log	ERROR: Array subscript out of range at line 30 column 5.
Drill sergeant	ERROR: You are outside of the sector of your array. Get back here now!
Strict nun school teacher	ERROR: Is pride the reason you think you can go beyond the range of your array: 10 Hail Marys.
Don't mess with me judo lady	ERROR: Your array subscript is out of range, fix it or I'll fix you.
Short and sweet	ERROR: Your array subscript is wrong. Please try again.
Voice of the heavens	ERROR: Thee have strayed from the confines of the defined array.
British lady	ERROR: This array subscript is rubbish.
Redneck guy	ERROR: Like a squirrel in a coon trap, you can't go outside your array.
Valley girl	ERROR: Like, your array is seriously grody to the max, whatevuh.
Yoda	ERROR: Of range at line 30 column 5 array subscript out. Yeesssssss.

Table 9. GPS log interpretation for an array log message

MACRO ERRORS

Macro coding can be complex and macro errors are sometimes cryptic at best. In the log below, the error message tells us that we are calling a macro with more parameters than we had defined. We need to either limit the macro call to the number of parameter defined in the %MACRO statement, or we should add another parameter to the macro. In this current state the macro will not be able resolve until the error is fixed. It should be noted that helpful options in debugging macros are MERROR, SERROR, MLOGIC, MPRINT, and SYMBOLGEN, and only the first two options are turned on by default.

```

33  %MACRO define(dsn,num);
34
35  DATA array; SET &dsn;
36  ARRAY nmiss(&num) a b c;
37  DO i=1 TO &num;
38    IF nmiss(i)=. THEN flag=1;
39  END;
40  RUN;
41
42  %MEND;
43
44  %define(new,3,'missing');
```

ERROR: More positional parameters found than defined.

Character	Log Message
SAS Log	ERROR: More positional parameters found than defined.
Drill sergeant	ERROR: You have more parameters than you defined. Your MACRO will fail. We do not fail here!
Strict nun school teacher	ERROR: Don't be greedy, you have more parameters than you need.
Don't mess with me judo lady	ERROR: One fundamental of Judo is balance, now balance your parameters.
Short and sweet	ERROR: Your MACRO call has too many parameters.
Voice of the heavens	ERROR: Thy parameters are irreconcilable.
British lady	ERROR: Your parameters are uncalled for, you have a bit too many.
Redneck guy	ERROR: Your parameters are all out of whack. You need give that a little looky loo.
Valley girl	ERROR: Like you have too many parameters, fer shur.
Yoda	ERROR: Positional parameters more than, found as defined.

Table 10. GPS log interpretation for a macro log message

CONCLUSION

The SAS log is your best ally for verifying that your code is doing what you think it is doing; however, understanding the messages in the SAS log can be a daunting and frustrating task. It can make the sanest people crazy. Listening to log messages in a different way could make all the difference. Our hope is to provide some fun and entertaining alternative interpretations of the notes, warnings, and errors that commonly appear in the SAS log. Hopefully, you will never look at the SAS log in the same way again.

REFERENCES

- Ainge, Matthew. "The Yoda-Speak Generator." September 2013. Available at <http://www.yodaspeak.co.uk/index.php>.
- Delwiche, Lora D. and Slaughter, Susan J. (2012). *The Little SAS Book: A Primer, Fifth Edition*, North Carolina: SAS Institute Inc.
- Mullin, Charley and Russell, Kevin. 2012. "Why Does SAS® Say That? What Common DATA Step and Macro Messages Are Trying to Tell You" *Proceedings of the SAS Global Forum Conference 2012*. Cary, NC: SAS. Available at: <http://support.sas.com/resources/papers/proceedings12/248-2012.pdf>.
- Slaughter, Susan J. and Delwiche, Lora D. 1997. "Errors, Warnings, and Notes (Oh My) A Practical Guide to Debugging SAS Programs." *Proceedings of the SAS Users Group International Conference*. Cary, NC: SAS. Available at: <http://www2.sas.com/proceedings/sugi22/BEGTUTOR/PAPER68.PDF>.

ACKNOWLEDGMENTS

Cast of characters who helped with voice-overs for the conference presentation are listed below. Thank you!

Character	Name
Drill sergeant	Renato Villacorte
Strict nun school teacher	Sue Douglass
Don't mess with me judo lady	AnnMaria DeMars
Short and sweet	Lora Delwiche
Voice of the heavens	David Pasta
British lady	Alexandra Sutton
Redneck guy	Ethan Miller
Valley girl	Rebecca Ottesen
Yoda	Jeff Sutton

CONTACT INFORMATION

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

Name: Rebecca Ottesen
Address: 1 Grand Avenue, Statistics Dept
City, State ZIP: San Luis Obispo, CA 93407-0405
Work Phone: 805-756-2709
Fax: 805-756-2700
E-mail: rottesen@calpoly.edu
Web: <http://statweb.calpoly.edu/rottesen/>

Name: Ethan Miller
Address: SRI International
333 Ravenswood Ave
City, State ZIP: Menlo Park, CA 94025
Work Phone: 650-859-6393
E-mail: ethan.miller@sri.com

DISCLAIMER: This paper is in no way sponsored or endorsed by: George Lucas, Lucasfilm Ltd., LucasArts Entertainment Co., or any affiliates. Star Wars and all its characters are © and TM Lucasfilm Ltd.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.