

An Efficient Way to Check Log Issue in SAS Enterprise Guide

Sichan Tang, Beigene, Shanghai

Dong Guo, Beigene, Shanghai

ABSTRACT

SAS Enterprise Guide (EG) is well-known by its powerful point-and-click interface and service-based options that help end-users to quickly and easily analyze data. Comparing with general SAS software, SAS Enterprise Guide summarizes the “Errors” and “Warnings” message in Log Summary Window automatically. It’s helpful for SAS log check. However, except “Errors” and “Warnings”, some special notes are also not allowed in SAS log in many big companies. In this paper, we share a method to highlight user-defined notes as “Warnings” in Log Summary Window and show you the benefit of this method.

INTRODUCTION

SAS® Enterprise Guide® - Your Gateway to SAS. It was originally promoted as a code generator for non-programmers. The truth is, however, that SAS Enterprise Guide has always allowed programmers to write their own code. In addition, it offers many features that are not included in PC SAS®.

We introduce some of the top Enterprise Guide features in the following areas:

- Using autocomplete
- Formatting programs
- Creating a process flow from a program
- Viewing the log and log summary

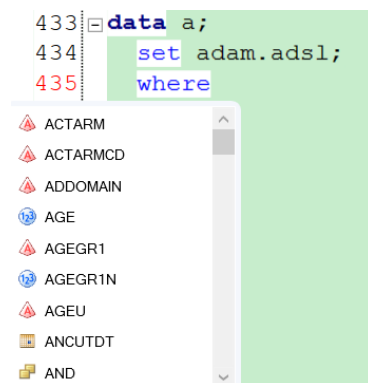
The Log Summary window lists how many errors, how many warnings, and how many notes are in the log. However, except “Errors” and “Warnings”, some special notes are also not allowed in SAS log in many big companies. We give a solution to define some special “Notes” as “Warning” in Log Summary and show you the benefit of this method.

BEING AN EFFICIENT PROGRAMMER

USING AUTOCOMPLETE

Enterprise Guide offers a point and click environment to constructing your code. It adds the autocomplete feature to SAS editor. The editor can suggest SAS statements /procedures /macro programs /macro variables /functions /formats /librefs /SAS data set names /SAS variable names.

For example, when you are specifying a variable name, such as on a WHERE statement, Enterprise Guide will display the list of variables in that data set by alphabetically. Moreover, EG will display the type for each variable. The blue circle with the 123 inside indicates that is a numeric variable. The red triangle with the A inside indicates it is a character variable. How much saving your time if you instantly know the variables spelling and its attribute.



Display 1. Enterprise Guide autocomplete variable name and type

<Paper title>, continued

In addition, if you hover over a keyword, you will see the basic syntax for that keyword as well as the description. The inline help also include links to product documentation, samples and SAS Notes, as well as relevant papers.

And the inline help is context sensitive. For example, if you hover over the VAR statement in a PROC PRINT step, the help is different than the VAR statement in a PROC MEANS step. Having that context sensitive help saves countless searches trying to get just the exact information you want online.

FORMATTING PROGRAMS

When you take over a code from a fresher in the half way, you may be uncomfortable with the format. But in Enterprise Guide, you can fix it, just by clicking edit -> format code or pressing CTRL + I.

For example, you inherit a code like that

```
data a;  
  set adam.adsl; where trt01pn=1; run;
```

Just pressing CTRL + I then the newly formatted code would be like that

```
data a;  
    set adam.adsl;  
    where trt01pn=1;  
run;
```

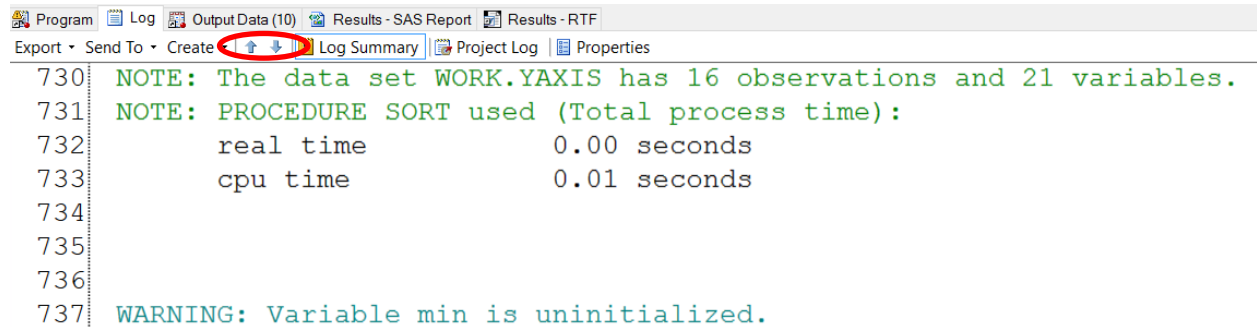
Moreover, you can change the indent space as you like. Click on Program -> Editor Options, then on the General tab, change the tab size from 4 to the number of spaces that you prefer.

CREATING A PROCESS FLOW FROM A PROGRAM

Usually ADAM code maybe maintained in a long term so code become long and complex, we were unsure if a data set created was being used elsewhere in the program. Enterprise Guide has a fantastic approach that runs through the program and identifies all the inputs and outputs for each step, so you can identify which created datasets that are never used. To create the process flow, click program -> Analyze -> Analyze for Program Flow. And you can see a visual process flow, then you can easily identify the inputs and outputs in each step.

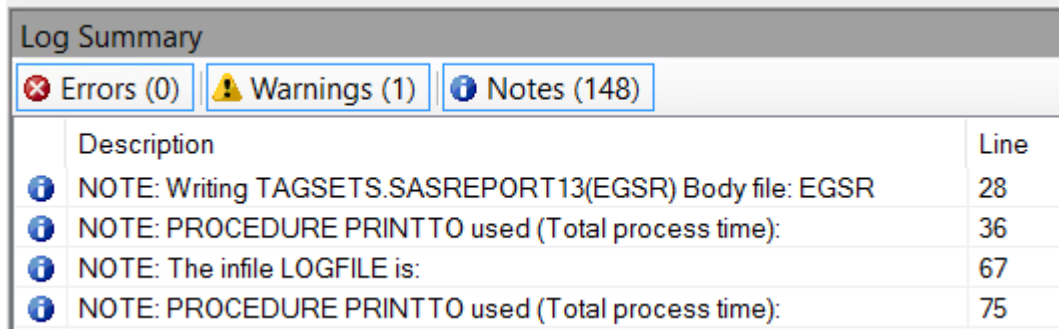
VIEWING THE LOG AND LOG SUMMARY

As every good SAS programmer knows, you should always, always, always check the log for error messages, warning messages, and notes that look unusual. In Enterprise Guide, you have two ways to review the log. One is manually navigating on the log tab. You can use arrows on the Log tab to quickly navigate to the next warning or error.



Display 2. log tab

The other way is view the log summary window. The Log Summary window lists how many errors, how many warnings, and how many notes are in the log.



Log Summary		
Errors (0) Warnings (1) Notes (148)		
	Description	Line
NOTE:	Writing TAGSETS.SASREPORT13(EGSR) Body file: EGSR	28
NOTE:	PROCEDURE PRINTTO used (Total process time):	36
NOTE:	The infile LOGFILE is:	67
NOTE:	PROCEDURE PRINTTO used (Total process time):	75

Display 3. log summary window

The Errors, Warnings, and Notes tabs are all toggles. If you click on the Notes tab that will hide the notes so that you can concentrate on errors and warnings. Clicking the Notes tab again will unhide the notes.

Clicking on one individual Note, Warning, or Error message will take you to that part of the log where that message occurred. You can more easily and quickly identify the problems in your program.

USER-DEFINING SPECIAL “NOTES” AS “WARNINGS” IN LOG SUMMARY WINDOW

Like “Errors” and “Warnings”, some special notes are also not allowed in SAS log in many big companies, such as “NOTE: Variable XX is uninitialized”, “NOTE: MERGE statement has more than one data set”, “NOTE: Numeric values have been converted to character values”.

Usually those special notes are drowned out by the mass of notes, so it’s time-consuming to find them out from reviewing log tab. Even using log summary window, it takes much time too since you need click note one by one.

Here we give a solution to fix it by replacing those special notes with warnings and directly showing them as “Warnings” in log summary window. Specifically, we insert one custom SAS codes after submitted code, so when we run own submitted code, the inserted code would run automatically, then those define special notes would be showed as “Warnings” in log summary window if they exist.

So what’s the magical code? We post it as below. Firstly we generate the macro variable `__NOTE_CHECK__` to define special notes and separate them by “|”. Then use `prxchange` to find and replace special ERROR to “ERROR:”, such as find and replace “ERROR 1-1” to “ERROR:”. Next use `prxparse` to generate a template which begins with “NOTE:” and its key word is the custom defined special note. Then use `prxmatch` to check whether the current note can match the template or not. If matched, indicates the current note is the custom defined special note. Then use `prxchange` to replace the “NOTE:” to “WARNING:”. Above three functions are SAS regular expression.

<Paper title>, continued

```
/* Insert custom code after submitted code here */
proc printto;
run;

%let __NOTE_CHECK__ =is uninitialized|Invalid|MERGE statement has more than one
data set|values have been converted|Input data set is empty|W.D format|Missing
values were generated|Unknown|will be overwritten by|Division by zero|Format was to
small;

data _null_;
  infile LogFile truncover lrecl=30000;
  input;
  length content$30000;
  content=_infile_;

  /* Change ERROR \d-\d to ERROR: */
  content = prxchange('s/ERROR( +\d+-\d+: +)/ERROR:\1/o', 1,content);

  /* Change PRG-ERROR: to ERROR: for %m_adam_create special log*/
  content = prxchange('s/^PRG-ERROR:/ERROR:\1/o', 1,content);

  /* Replace special notes with warning*/
  pattern = prxparse("(?<=^NOTE:).+\b(&__NOTE_CHECK__)\b/o");
  if prxmatch(pattern, content) then do;
    content = prxchange('s/^NOTE: /WARNING: /o',1,strip(content));
  end;

  /* put log out*/
  len=length(content);
  putlog content $varying200. len;
run;

%symdel __NOTE_CHECK__;
```

Of course, firstly you need print log so you need insert below code before submitted code.

```
/* Insert custom code before submitted code here */
filename LogFile temp;
Proc printto log=LogFile new;
run;
```

click on Tools -> Options -> SAS Programs then you can insert above two custom codes before and after submitted code.

This is a big time-saver to search special note since it is showed as “Warnings” in log summary window, so you don’t need to find them out from the mass of “Notes”. You can also change macro variable __NOTE_CHECK__ to customize what you want to pay attention to, as they will be highlighted as “Warnings” or “Errors”.

CONCLUSION

Enterprise Guide provides a point and click access to the SAS® System, but more than that, it offers a place where users can customize code in their ways.

The log summary window is a useful window to quickly search problems in your program. By inserting our custom code, you can instantly find special notes which you want to concentrate on and as a result you will be a more efficient SAS programmer.

REFERENCES

M. Michelle Buchecker, ThotWave Technologies, LLC. “Kicking and Screaming Your Way to SAS® Enterprise Guide®”. SAS Forum, Paper 1265-2014.

<Paper title>, continued

Dave Shea, Skylark Limited, Wellington, New Zealand, "SAS® Enterprise Guide® - Your Gateway to SAS®". SAS Forum, Paper 1265-2014.

RECOMMENDED READING

The Little SAS Book for Enterprise Guide 4.2 - by Susan J. Slaughter and Lora D. Delwiche

CONTACT INFORMATION

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

Sichan Tang
Dong Guo

Email: sichan.tang@beigene.com
Dong.guo@beigene.com

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.