Paper SS01

A Simple Way to Improve the Efficiency of Log Checking in SAS Enterprise Guide

Danny Hsu, Seagen, Bothell, WA, USA

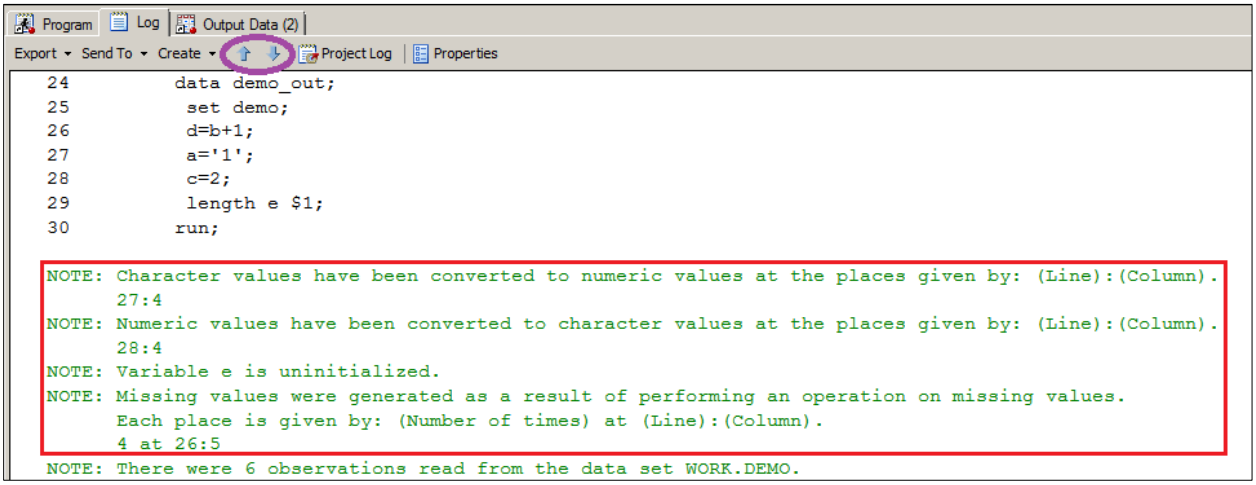
ABSTRACT

After each program run, SAS® Enterprise Guide® (SAS EG) generates a log displaying information regarding the process, which includes notes, warnings, and errors. Navigating through log messages in SAS EG to look for faulty messages could be a time-consuming and frustrating process. This article shares a simple way of utilizing the system options of SAS EG to implement the checklog codes. As the result of the checklog codes, a log summary will be provided after each program run and it displays a list of custom-defining log messages, including ERROR, WARNING, MISSING, …etc. The programmers could then refer to the log summary to quickly understand if any possible issues exist within their programs. This process would definitely help to improve the efficiency of the debugging process, hence, leading to a superior programming quality.

INTRODUCTION

To a SAS programmer, the SAS log file contains many bits of information such as errors, warnings, and notes that are critical to the diagnosis of the program code; therefore, the ability to efficiently gather this information is highly desirable and valuable to a programmer's productivity. SAS EG is a popular program development and execution tool. It highlights the errors and warnings, but some notes are also worthy of our attention. However, in SAS EG, it could be exhausting to go through a very long log to read all error/warning/note messages. For example, after each program run, programmers could use the arrows circled in *Display 1* to look for the errors and warnings, which can be a tedious and tiring process. In addition, some notes in the log file are just informational, but others can reveal some potential problems in the program code, which most companies or programmers do not allow to remain in their program logs. For example, the notes highlighted in *Display 1* below. Unfortunately, SAS EG does not distinguish such notes.

[Display 1 – Log tab in SAS EG]

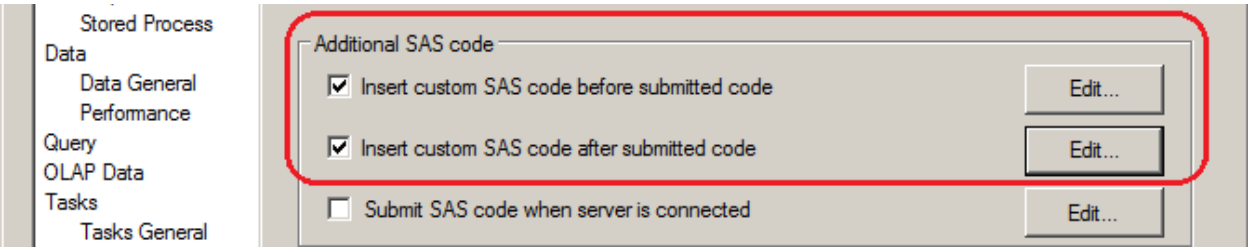


To address these issues, the author developed a method to improve the standard log checking in EG by utilizing SAS EG's ability to insert customized checklog code both before and after submitted code, therewith post-processing the log file and providing a customized summary of the error, warning, and pertinent note messages. The following highlights the key steps:

INSERT CHECKLOG CODE IN SYSTEM OPTIONS

A checklog tool with selected log criteria could be inserted into the system options highlighted in *Display 2* below. Sample custom checklog codes can be found in **Appendix 1** and **Appendix 2.**

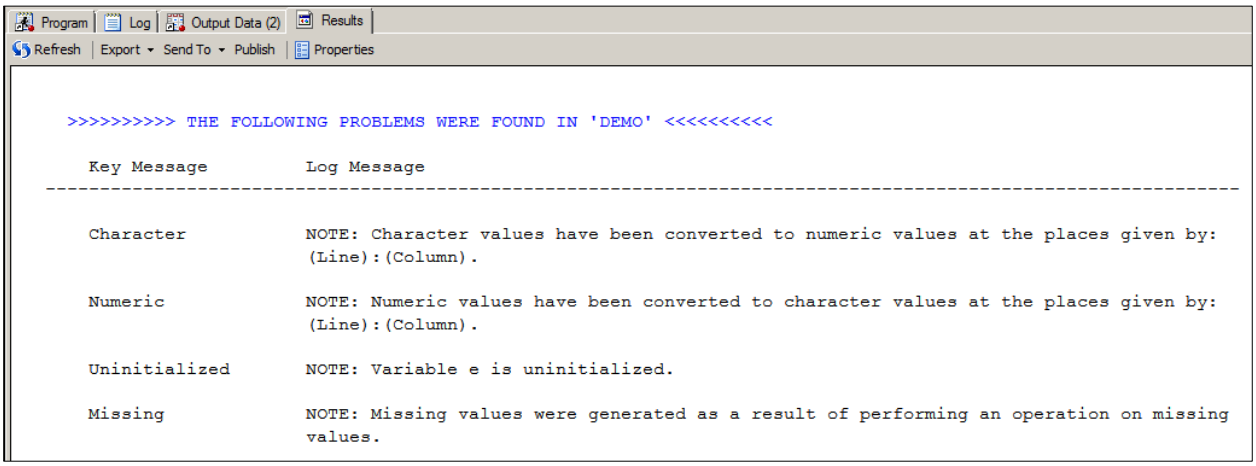
[Display 2 – Options in SAS EG]



LOG SUMMARY RESULT

After implementing these checklog code blocks before and after each program run, this checklog code suite will be executed automatically and a log report will be displayed in the Results tab to categorize the unwanted messages including “Error”, “Warning”, “Missing”, “Uninitialized”, ...etc. For instance, all 4 notes highlighted in *Display 1* are listed in *Display 3*. By directly referring to this log summary, programmers could quickly learn if any unwanted messages exist in their logs without having to manually search for them.

[Display 3 – Results tab in SAS EG with log summary result]



CONCLUSION

To look for unwanted messages in SAS EG is time consuming; therefore, an efficient and automated checklog tool is a game changer, saving a great deal of time for programmers while offering an effective way to improve programming quality. Thanks to this checklog tool, we will have more time for coffee breaks!

REFERENCES

https://support.sas.com/resources/papers/proceedings12/098-2012.pdf
https://support.sas.com/resources/papers/proceedings/proceedings/sugi27/p096-27.pdf
https://support.sas.com/resources/papers/proceedings17/1173-2017.pdf

ACKNOWLEDGMENTS

Thanks to Wei Qian for reviewing the paper and providing additional insight.

CONTACT INFORMATION

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

Danny Hsu
Seagen Inc.
21717 30th Dr SE
Bothell, WA 98021
Email: shhsu@seagen.com
Web: https://www.seagen.com

APPENDIX 1

Code to inset into the option, “Insert custom SAS code before submitted code”

```
filename log_ temp;
proc printto log=log_ new;
run;
```

APPENDIX 2

Code to inset into the option, “Insert custom SAS code after submitted code”

```
proc printto;
run;

data _null_;
    infile logfile;
    input;
    putlog _infile_;
run;

data _log;
    infile logfile delimiter='00'x MISSOVER DSD lrecl=32767;
    firstobs=1;
    format key $100. logtext $1000.;
    input  logtext $;
    if logtext=:'ERROR' then key='Error';
    else if logtext=:'WARNING' then key='Warning';
    else if upcase(logtext)=:'NOTE: CHARACTER' then key='Character';
    else if upcase(logtext)=:'NOTE: NUMERIC' then key='Numeric';
    else if upcase(logtext)=:'NOTE: MISSING' then key='Missing';
    else if upcase(logtext)=:'NOTE: VARIABLE' then key='Note: Variable';
    if key^='';
    n=_n_;
run;

proc report data=_log headline;
    column n key logtext;
    define n      / order noprint;
    define key    / display "Key Message" width=20;
    define logtext/ display "Log Message" width=110;
    break before n/skip;
    title "THE FOLLOWING MESSAGES WERE FOUND:";
run;
```