

Instant Interactive SAS® Log Window Analyzer

Palanisamy Mohan, ICON Clinical Research India Pvt Ltd
Amarnath Vijayarangan, Emmes Services Pvt Ltd, India

ABSTRACT

An interactive SAS® environment is preferred for developing programs as it gives the flexibility of instantly viewing programming log in the Log window. The programmer must review the log window to ensure that each and every single line of a written program is running successfully without displaying any messages defined by SAS that are potential errors. The number of lines generated in the log window varies depending on the complexity of the program. Reviewing the log window every time is not only time consuming but also prone to manual error for any level of programmer. Even to confirm that the log is free from error, the programmer must check the log. Currently in the interactive SAS environment there is no way to get an instant notification about the generated log from the Log window, indicating whether there have been any messages defined by SAS that are potential errors. This paper introduces an instant approach to analyze the Log window using the SAS macro %ICHECK that displays the report instantly in the same SAS environment.

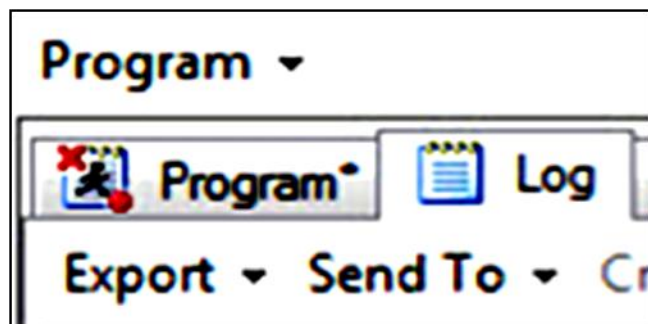
INTRODUCTION

The successful programming development is ensured when the log is free from the potential messages. Updating the programs and reviewing the log in the Log window is a to and fro process. As the program grows, the number of lines reported in the log window increases and becomes time consuming task to go through the log. There are plenty of concepts to analyze the log when they are available as a file but this paper introduces a new approach to analyze the log generated in the Log window rather than a file. The SAS macro %ICHECK generates a report in a separate window using %WINDOW and DM statement with the summary of all the SAS defined messages that are potential to error. The programmer does not have to add %ICHECK at the end of the program. Whether a single statement, DATA step, PROC step, or the whole program is submitted, the %ICHECK macro is automatically executed at the end of every submission. It might be surprising to you how a compiled macro can be executed without calling them in the Editor window. But it is possible with the %ICHECK and is developed using Base SAS product.

SIMILAR FUNCTIONALITY

SAS Enterprise Guide

When a program is submitted in SAS Enterprise Guide, the program editor tab shows X mark if there are any errors in the execution as shown below and red mark gets your attention immediately.



Display 1. SAS Enterprise Guide Editor and Log Tabs

SAS Studio

SAS Studio is a developmental web application for SAS that you access through your web browser. It also has the functionality of summarizing the log messages as follows:



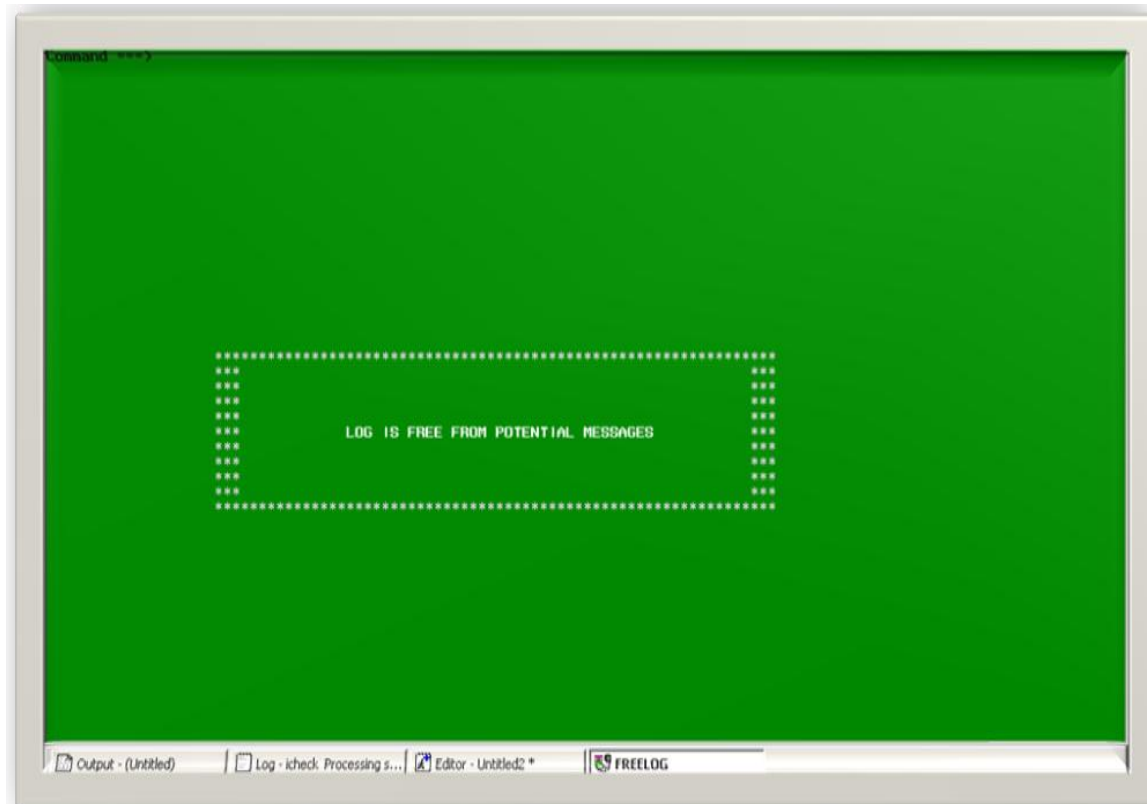
Display 2. SAS Studio Editor and Log tabs

Base SAS

Currently there is no option in Base SAS to indicate the execution status like SAS Enterprise Guide or provides the summary of the log like SAS Studio. The proposed approach combines the functionality of SAS Enterprise Guide and SAS Studio using the %ICHECK macro.

The %ICHECK generates the reports in two types of custom windows.

When the log is free from potential messages, the report window named as FREELOG is displayed in green color with the message "LOG IS FREE FROM POTENTIAL MESSAGES"



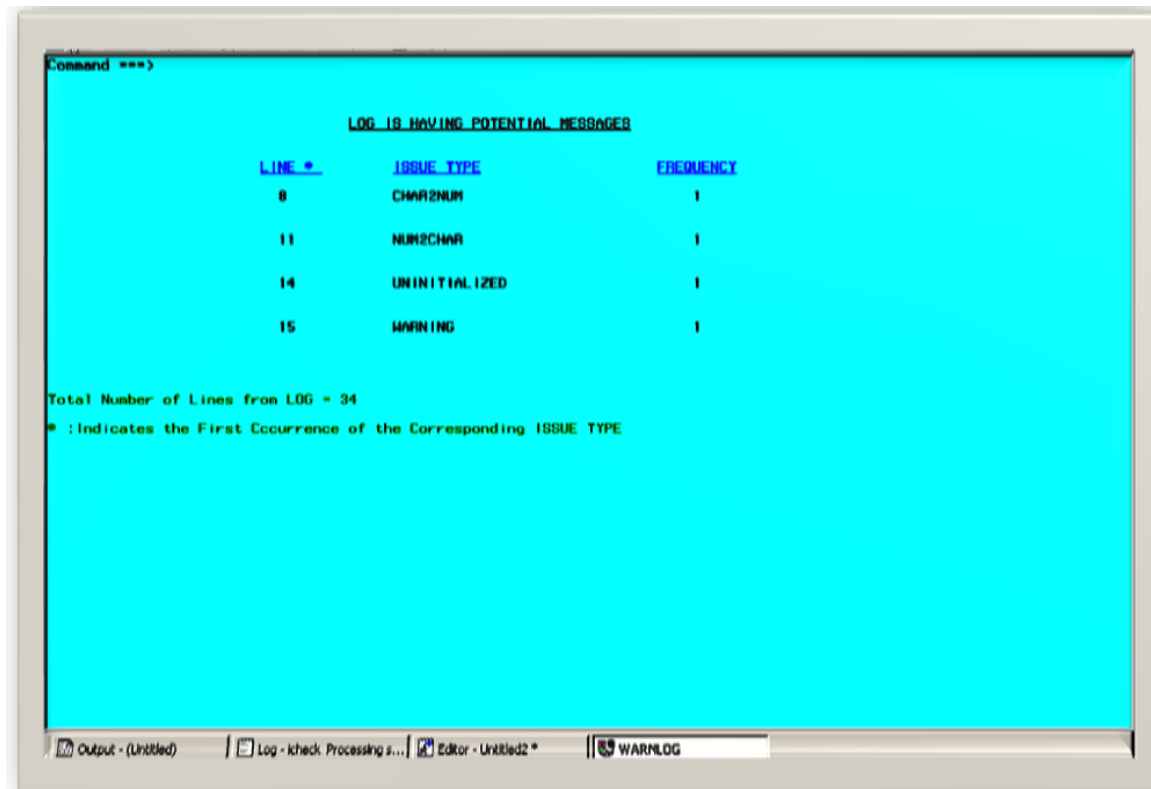
Display 3. FREELOG Window

The second type of window is generated as follows with the name WARNLOG in greenish-blue color when there is any potential messages occurred in the log.

The WARNLOG window has the following information:

- LINE #: The first occurrence of the corresponding issue type
- ISSUE TYPE: Provides the name of the potential messages in a short form which is self explainable
- FREQUENCY: How many times the corresponding issue type occurred in the log

In addition to the above three columns there are two footnotes provided at the right bottom of the report. It provides the total number of lines in the log. This can be connected with the LINE # column to get the sense of the first occurrence of the corresponding issue type.

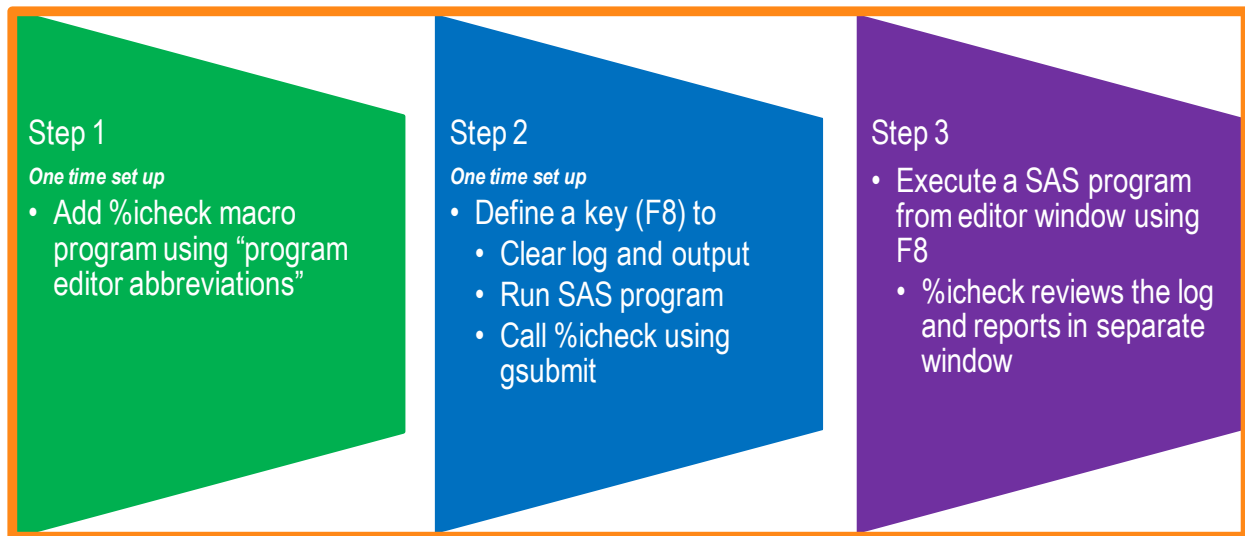


Display 4. WARNLOG Window

SETTING %ICHECK TO RUN AT THE END OF EVERY EXECUTION

While developing the programs adding %ICHECK macro call at the end of the every submission is not easy and not advisable to do. This was the challenge initially when the %ICHECK macro was developed. Later, this problem was addressed by making use of Editor Abbreviations and Key definitions.

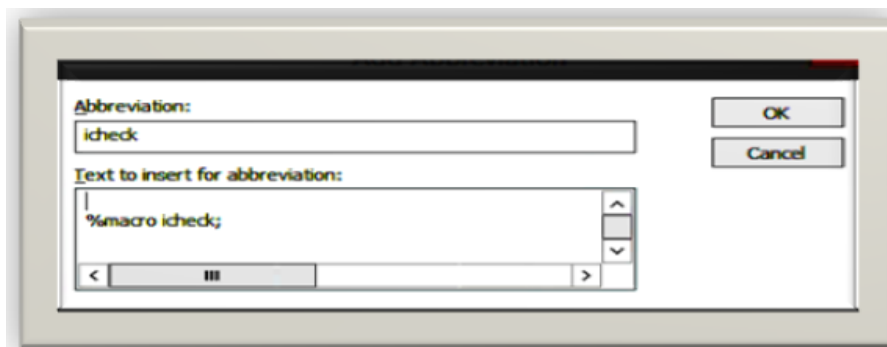
The process flow to set up %ICHECK is explained as follows.



Display 5. Process Flow

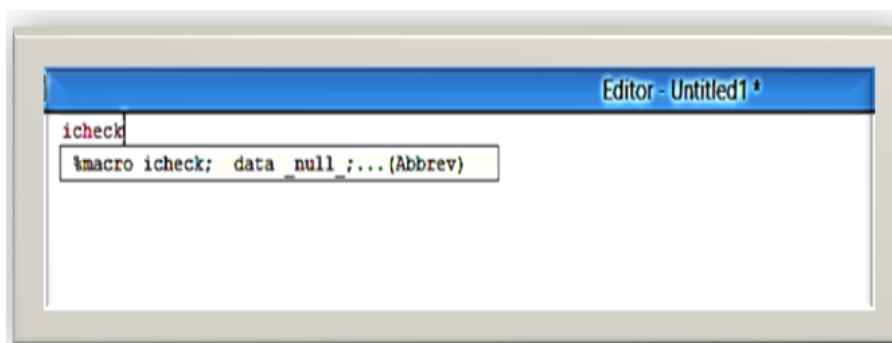
SAS PROGRAM EDITOR ABBREVIATIONS

The proposed approach utilizes the Editor Abbreviations method to quickly get the %ICHECK macro in the editor window. It is a method to retrieve the text(s) that are stored using a particular keyword. Otherwise, every time it requires some effort to compile the macro from a folder or to add it in setup/auto execution file if allowed. The macro %ICHECK is stored with a keyword icheck as shown below:



Display 6. Add Abbreviation

The %ICHECK macro can be easily called in Editor Window and compiled with keyword icheck as shown below:



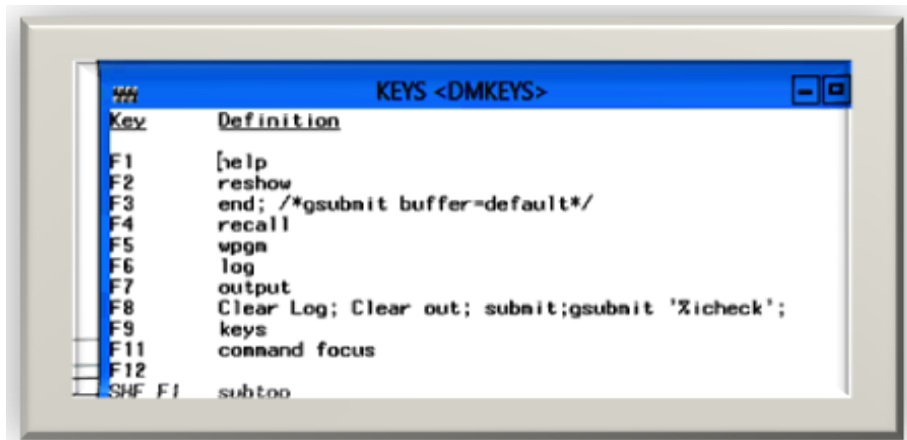
Display 7. Macro Call %icheck

KEY DEFINITION

SAS provides plenty of options to customize the shortcut keys to ease your work. Use any desired key and assign the below commands:

Clear Log; Clear Out; submit; gsubmit '%icheck';

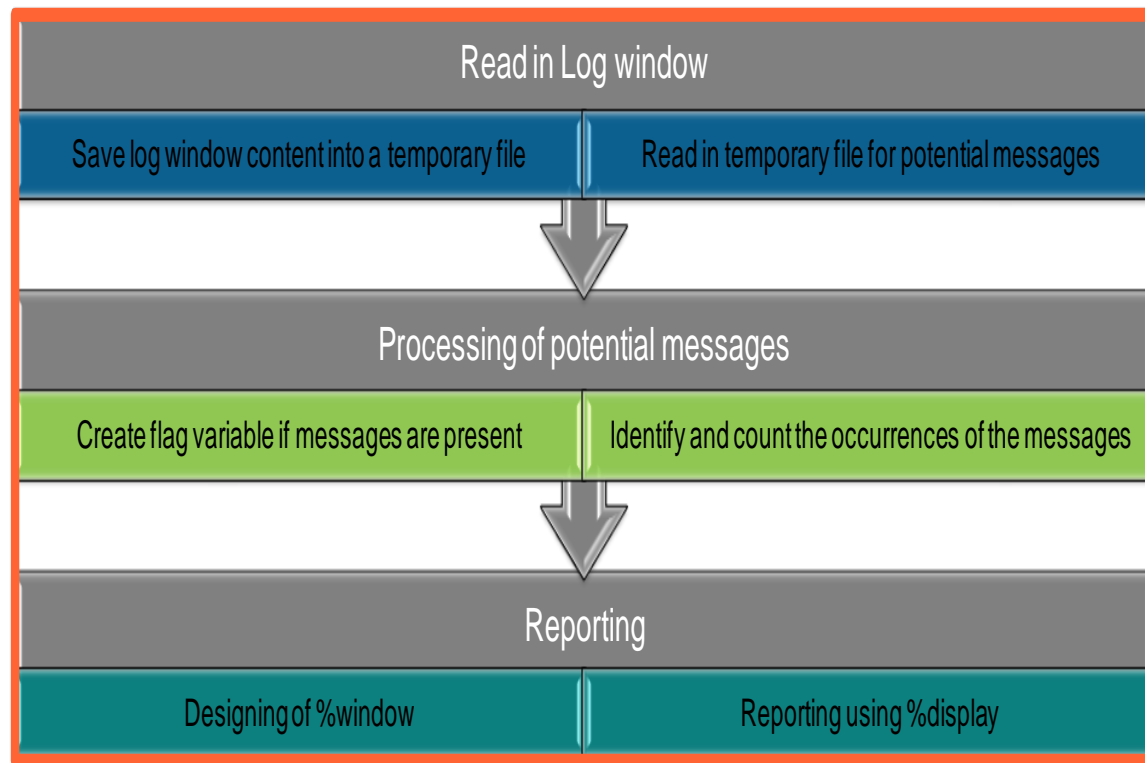
The command GSUBMIT is used to submit SAS code stored on the clipboard. For the illustration purpose, F8 key is used as shown below.



Display 8. Display Manager Keys

%CHECK MACRO

The skeleton of the %ICHECK macro is explained as follows in a simple way.



Display 9. Execution flow

CONCLUSION

The proposed solution is developed only using Base SAS product and SAS programming. It can be used in a Windows or UNIX interactive SAS environment without requiring any user inputs. With the proposed approach, there is a significant benefit in the log review process and 100% time saving when log is free from potential messages and 0% chances for any potential messages to be over looked for all levels of programmers. Similar functionality can also be introduced in the Base SAS product itself.

CONTACT INFORMATION

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

Palanisamy Mohan,
ICON Clinical Research India Pvt Ltd,
Palanisamy.Mohan@iconplc.com
palanisas@gmail.com

Amarnath Vijayarangan
Emmes Services Pvt Ltd, Bangalore, India
avijayarangan@emmes.com
amarnath7@gmail.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.

APPENDIX

%macro icheck;

***Identify the path of WORK library ;
%let wpath= %sysfunc(pathname(work));

dm 'log; file "&wpath/ichack.log" replace';
dm "log; ";

```
data ichack;
  length IssueType txt $ 100;
  infile "&wpath/ichack.log" length=len lrecl=32000 ;
  input SystemMessage $varying2000. len ;
  IssueType='';
  SystemMessage=upcase(SystemMessage);
  if strip(substr(SystemMessage,1,5))='ERROR' then IssueType='ERROR';
  if strip(substr(SystemMessage,1,7))='WARNING' then IssueType='WARNING';
  if scan(SystemMessage,1,':')='NOTE' then do;
    if index(SystemMessage,'UNINITIALIZED') then IssueType='UNINITIALIZED';
    if index(SystemMessage,'MISSING') then IssueType='MISSING';
    if index(SystemMessage,'CHARACTER VALUES HAVE BEEN') then IssueType='CHAR2NUM';
    if index(SystemMessage,'NUMERIC VALUES HAVE BEEN') then IssueType='NUM2CHAR';
    if index(SystemMessage,'INVALID DATA FOR') then IssueType='INVALIDDATA';
    if index(SystemMessage,'MERGE STATEMENT HAS MORE THAN') then IssueType='MERGE';
    if index(SystemMessage,'NOTE: DIVISION BY ZERO DETECTED') then IssueType='DIVISIONBY0';
    if index(SystemMessage,'EXTRANEIOUS INFORMATION') then IssueType='EXTRANEIOUS';
    if index(SystemMessage,'W.D FORMAT') then IssueType='WDFORMAT';
    if index(SystemMessage,'REPEATS OF BY VALUES') then IssueType='REPEATS';
    if index(SystemMessage,'MATHEMATICAL OPERATIONS COULD NOT') then IssueType='MATHSOPER';
    if index(SystemMessage,'INTERACTIVITY DISABLED WITH') then IssueType='INTERACTIVITY';
    if index(SystemMessage,'NO OBSERVATION') then IssueType='NOOBSERVATION';
    if index(SystemMessage,'LOST CARD') then IssueType='LOSTCARD';
    if index(SystemMessage,'SAS SYSTEM STOPPED PROCESSING') then IssueType='SASSTOPPED';
  end;
  txt=strip(IssueType);
  lineno+1;
  call symputx('lastlineno',lineno);
  if IssueType^='';
  keep IssueType lineno SystemMessage Txt ;
run;
```

```
proc sort data=ichack out=FirstOccur;
  by IssueType lineno;
run;
```

```
data FirstOccur;
  set FirstOccur;
  by IssueType lineno;
  if first.IssueType;
  keep IssueType lineno;
```

```

run;

*** Delete temporary files;
%macro fdel(file=);
  %let rc= %sysfunc(filename(fref,&file));
  %let rc= %sysfunc(fdelete(&fref));
%mend;

*** Delete the temporary created log file;
%fdel(file=&wpath/ichack.log);

proc freq data=ichack noprint;
  tables IssueType*txt /out=ichack_Cnt(drop=percent);
run;

data ichack_Cnt;
  merge ichack_Cnt FirstOccur;
  by IssueType;
run;

proc sort data=ichack_Cnt;
  by lineno;
run;

%let FreeLog=0;
data _null_;
  set ichack_Cnt;
  call symputx('FreeLog',_n_);
run;

%put &FreeLog ;

%if &FreeLog=0 %then %do;
  %window FreeLog color=green
  #20 @20 '*****' color=grey
  #21 @20 '***' color=grey
  #22 @20 '***' color=grey
  #23 @20 '***' color=grey
  #24 @20 '***' color=grey
  #25 @20 '***' color=grey @35 'LOG IS FREE FROM POTENTIAL MESSAGES' color=white @81 '***' color=grey
  #26 @20 '***' color=grey
  #27 @20 '***' color=grey
  #28 @20 '***' color=grey
  #29 @20 '***' color=grey
  #30 @20 '*****' color=grey
  ;
  %display FreeLog ;
%end;

%if &FreeLog^=0 %then %do;
  data test;
    set ichack_Cnt end=eol;
    retain id 6;
    id+3;
    string="#||strip(put(id,best.)) || " @27 ' ' || cats(lineno) ||
      " " @40 ' ' || strip(TXT) || " ' #||strip(put(id,best.)) ||
      " @74 ' ' || " " || cats(count) ||'";
    file "&wpath/ichack.txt";
    if _n_=1 then do;
      put '%window WarnLog color=CYAN';
      put '#4 @35 "LOG IS HAVING POTENTIAL MESSAGES" a=highlight color=red a=underline';
      put '#7 @25 "LINE # " a=highlight color=BLUE a=underline';
      put '#7 @40 "ISSUE TYPE" a=highlight color=BLUE a=underline';
      put '#7 @70 "FREQUENCY" a=highlight color=BLUE a=underline';
    end;
    put string 'a=highlight color=black';
    if eol then do;
      string="#||strip(put(id+5,best.)) ||
        " @1 'Total Number of Lines from LOG = &lastlineno' a=highlight color=green";
      put string;
      string="#||strip(put(id+7,best.)) ||
        " @1 '# :Indicates the First Occurrence of the Corresponding ISSUE TYPE' a=highlight color=green";
      put string;
      put ';';
      put '%display WarnLog;';
    end;
  run;
  %include "&wpath/ichack.txt";
  %fdel(file=&wpath/ichack.txt);
%end;
%mend ichack;

```