PharmaSUG China 2014 - Paper CC08

Automated Log Analyzer Dashboard

Palanisamy Mohan, Cognizant Technology Solutions, India

Amarnath Vijayarangan, Genpact, India

ABSTRACT

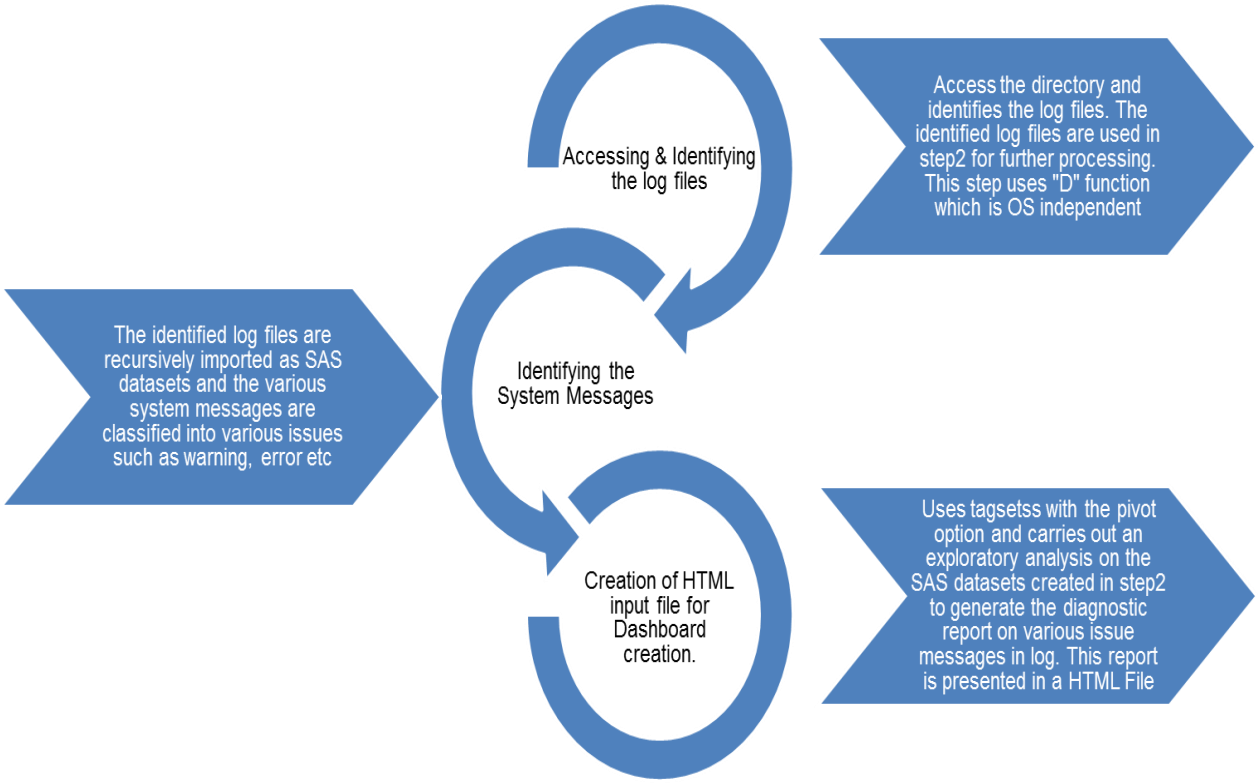
The importance of validating a SAS® program through the generated log file is inevitable. A successful execution would require an ERROR, WARNING and other system message free log. Though, the severity of NOTE or WARNING might not be very high, but there are chances for multiple NOTES or WARNINGS together in a program that can cause severe problems or incorrect results equal to an error message. This means that one needs to carefully review all the logs that have been generated when a program is executed, but a manual review of 1000 + logs becomes taxing and time consuming. To facilitate this we propose a very simple and a graphical solution which would enable any level programmer not to miss out on any system messages that gets generated when N number of programs is executed.

INTRODUCTION

The SAS programs in various domains generate very large and N number of log files when executed. For instance, Clinical Research domain demands 100 + SAS programs to be either executed in a batch mode or interactive mode for a final delivery, which is validated manually. This means a programmer needs to review 1000 + lines of code in the multiple logs manually where certain seemingly unimportant messages might be overlooked and also a manual review is really time consuming process.

The automated log analyzer dashboard will help in a 3600 degree review of the generated SAS Logs; a program is developed to ensure that no system generated message is overlooked. The automated log analyzer scans each and every single line of the entire log file in the directory for any system generated messages and provides a visual report: providing an overall summary on various system generated messages by SAS logs.

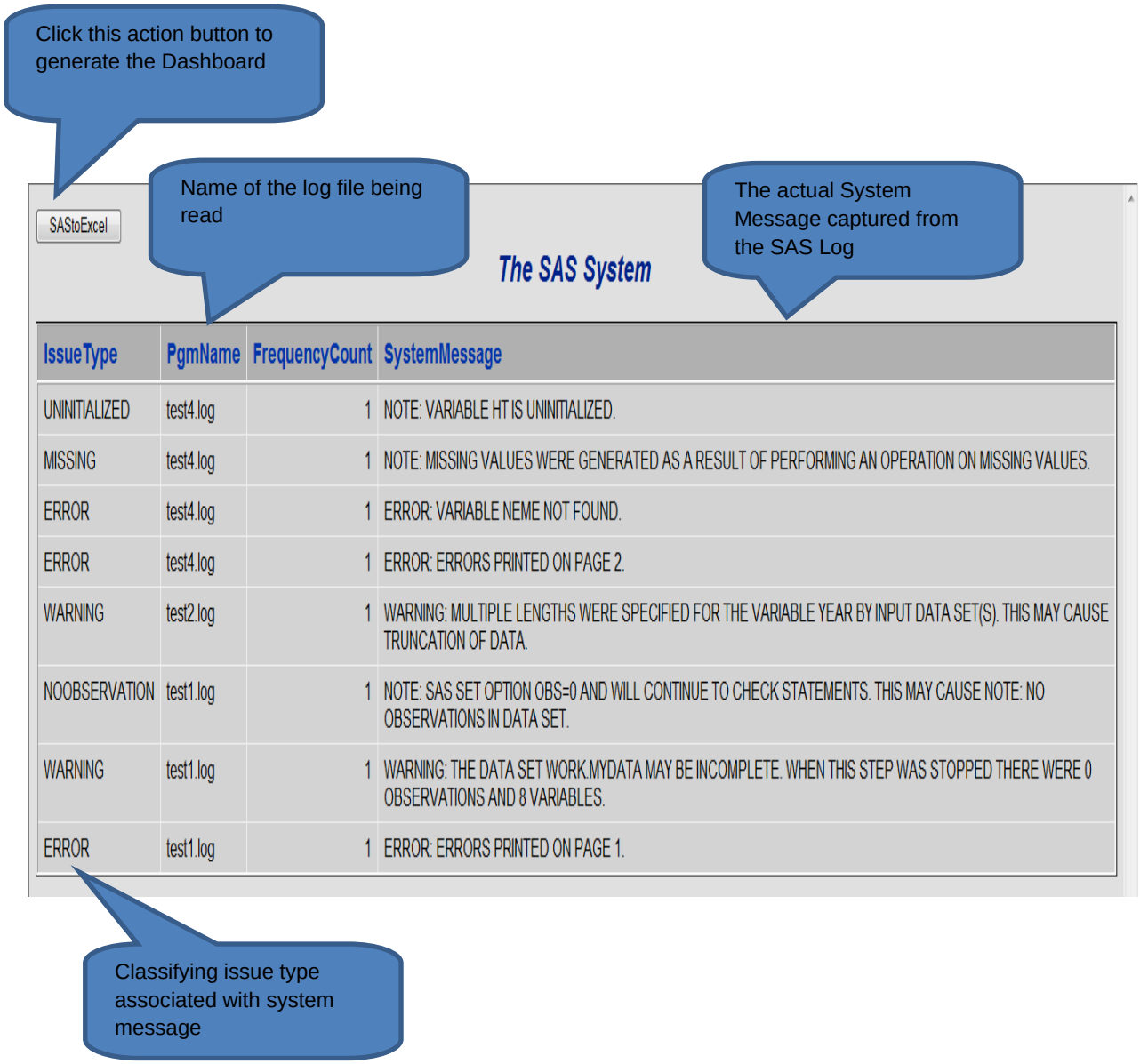
SAS PROGRAM FLOW



Once the HTML files are generated in the final step, these HTML files are supplied in as input to generate the dashboard as shown in the below section.

DASHBOARD GENERATION USING HTML FILE

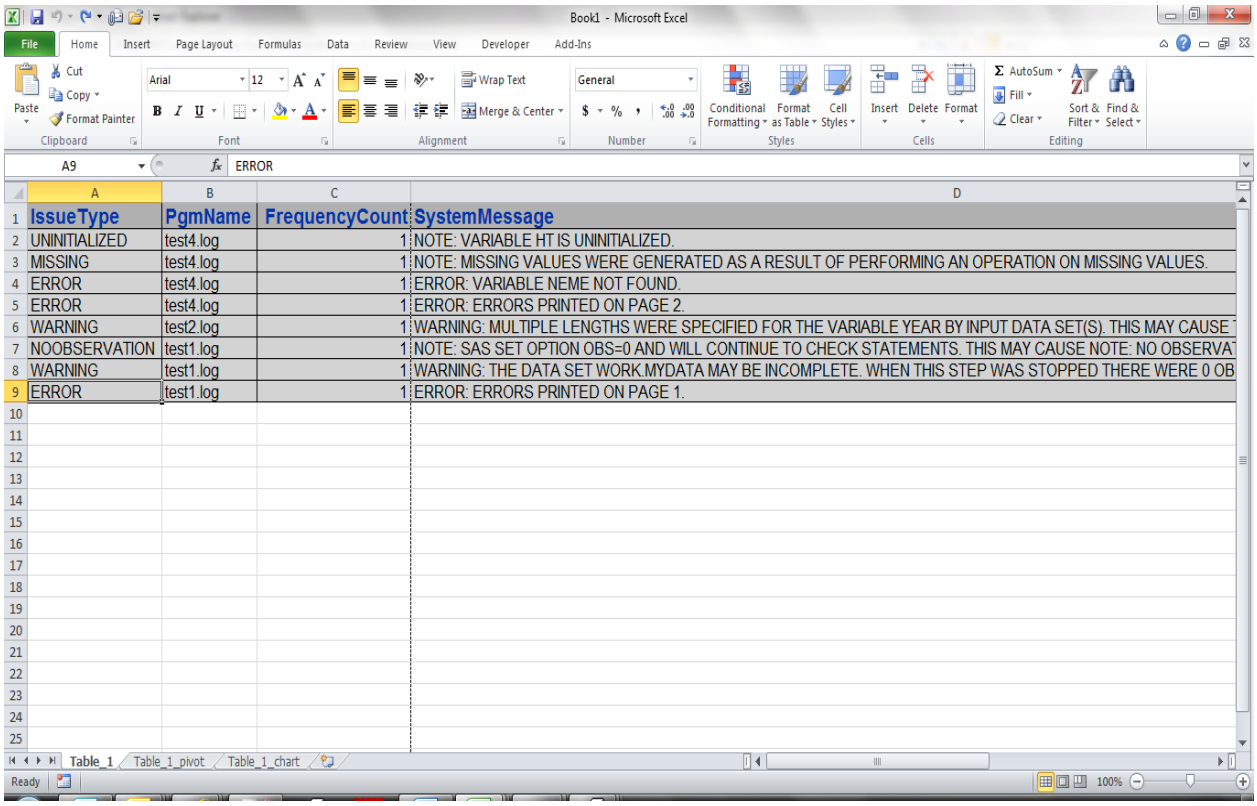
Using the HTML file we create the Dashboard to present the summary of various system messages.



IssueType	PgmName	FrequencyCount	SystemMessage
UNINITIALIZED	test4.log	1	NOTE: VARIABLE HT IS UNINITIALIZED.
MISSING	test4.log	1	NOTE: MISSING VALUES WERE GENERATED AS A RESULT OF PERFORMING AN OPERATION ON MISSING VALUES.
ERROR	test4.log	1	ERROR: VARIABLE NEME NOT FOUND.
ERROR	test4.log	1	ERROR: ERRORS PRINTED ON PAGE 2.
WARNING	test2.log	1	WARNING: MULTIPLE LENGTHS WERE SPECIFIED FOR THE VARIABLE YEAR BY INPUT DATA SET(S). THIS MAY CAUSE TRUNCATION OF DATA.
NOOBSERVATION	test1.log	1	NOTE: SAS SET OPTION OBS=0 AND WILL CONTINUE TO CHECK STATEMENTS. THIS MAY CAUSE NOTE: NO OBSERVATIONS IN DATA SET.
WARNING	test1.log	1	WARNING: THE DATA SET WORK.MYDATA MAY BE INCOMPLETE. WHEN THIS STEP WAS STOPPED THERE WERE 0 OBSERVATIONS AND 8 VARIABLES.
ERROR	test1.log	1	ERROR: ERRORS PRINTED ON PAGE 1.

SAS to Excel action button shown in the above HTML file triggers the Dashboard with three tabs as shown in the following sections. The column FrequencyCount with the constant value 1 is used while creating charts and summary report using pivot table method.

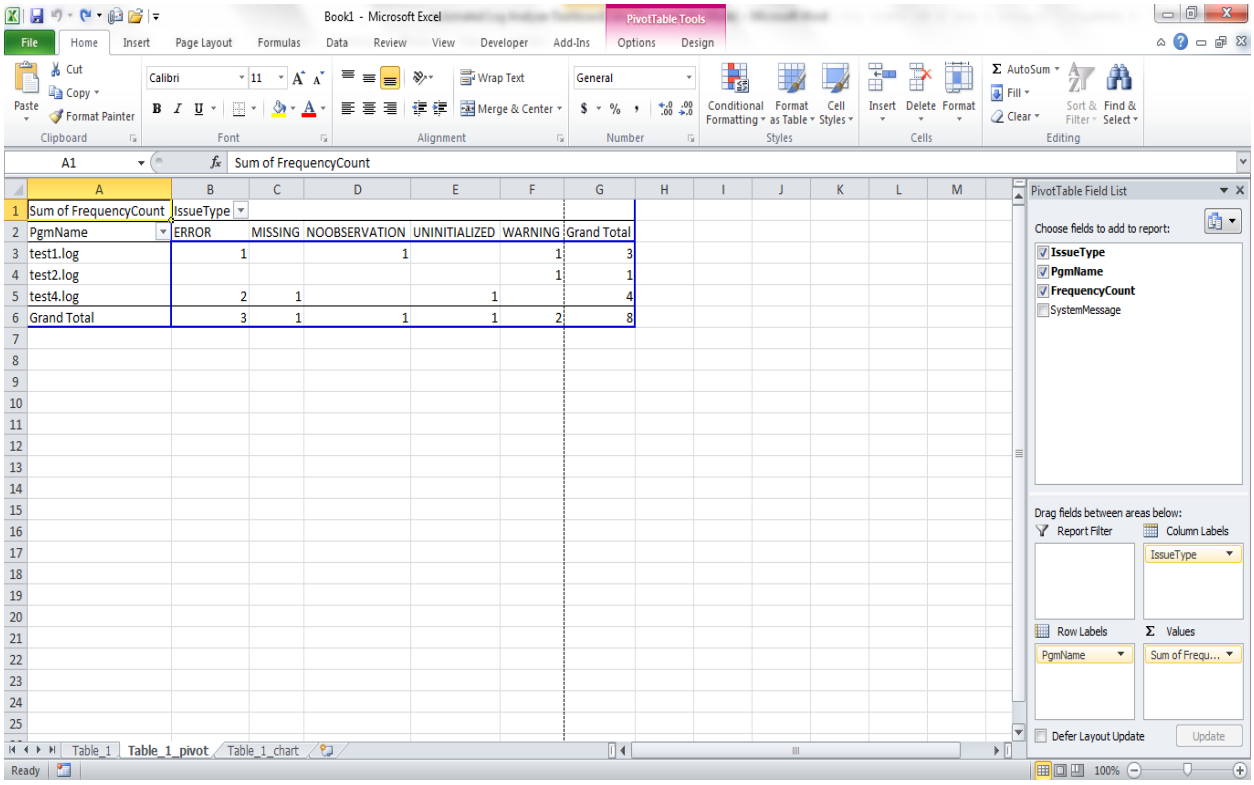
TAB1 - TABLE_1: ACTUAL SYSTEM MESSAGES FROM EACH SAS LOG FILE



	A	B	C	D
1	IssueType	PgmName	FrequencyCount	SystemMessage
2	UNINITIALIZED	test4.log	1	NOTE: VARIABLE HT IS UNINITIALIZED.
3	MISSING	test4.log	1	NOTE: MISSING VALUES WERE GENERATED AS A RESULT OF PERFORMING AN OPERATION ON MISSING VALUES.
4	ERROR	test4.log	1	ERROR: VARIABLE NEME NOT FOUND.
5	ERROR	test4.log	1	ERROR: ERRORS PRINTED ON PAGE 2.
6	WARNING	test2.log	1	WARNING: MULTIPLE LENGTHS WERE SPECIFIED FOR THE VARIABLE YEAR BY INPUT DATA SET(S). THIS MAY CAUSE
7	NOOBSERVATION	test1.log	1	NOTE: SAS SET OPTION OBS=0 AND WILL CONTINUE TO CHECK STATEMENTS. THIS MAY CAUSE NOTE: NO OBSERVA
8	WARNING	test1.log	1	WARNING: THE DATA SET WORK.MYDATA MAY BE INCOMPLETE. WHEN THIS STEP WAS STOPPED THERE WERE 0 OB
9	ERROR	test1.log	1	ERROR: ERRORS PRINTED ON PAGE 1.

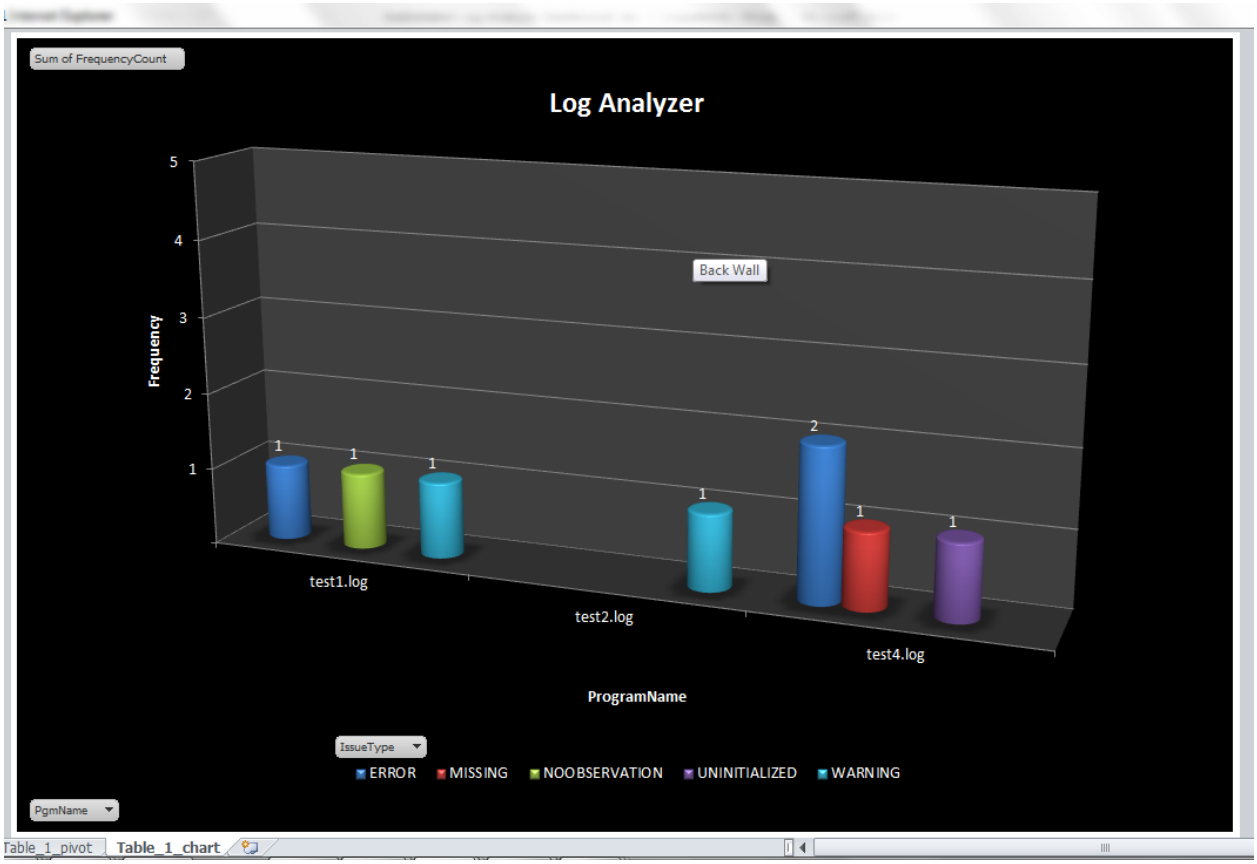
The first tab named Table_1 is a conversion of the HTML file generated in section 1.3 which contains the exhaustive list of all the system messages extracted from the SAS logs into an excel file.

TAB2 - TABLE_1_PIVOT: DESCRIPTIVE ANALYSIS OF THE SYSTEM MESSAGES



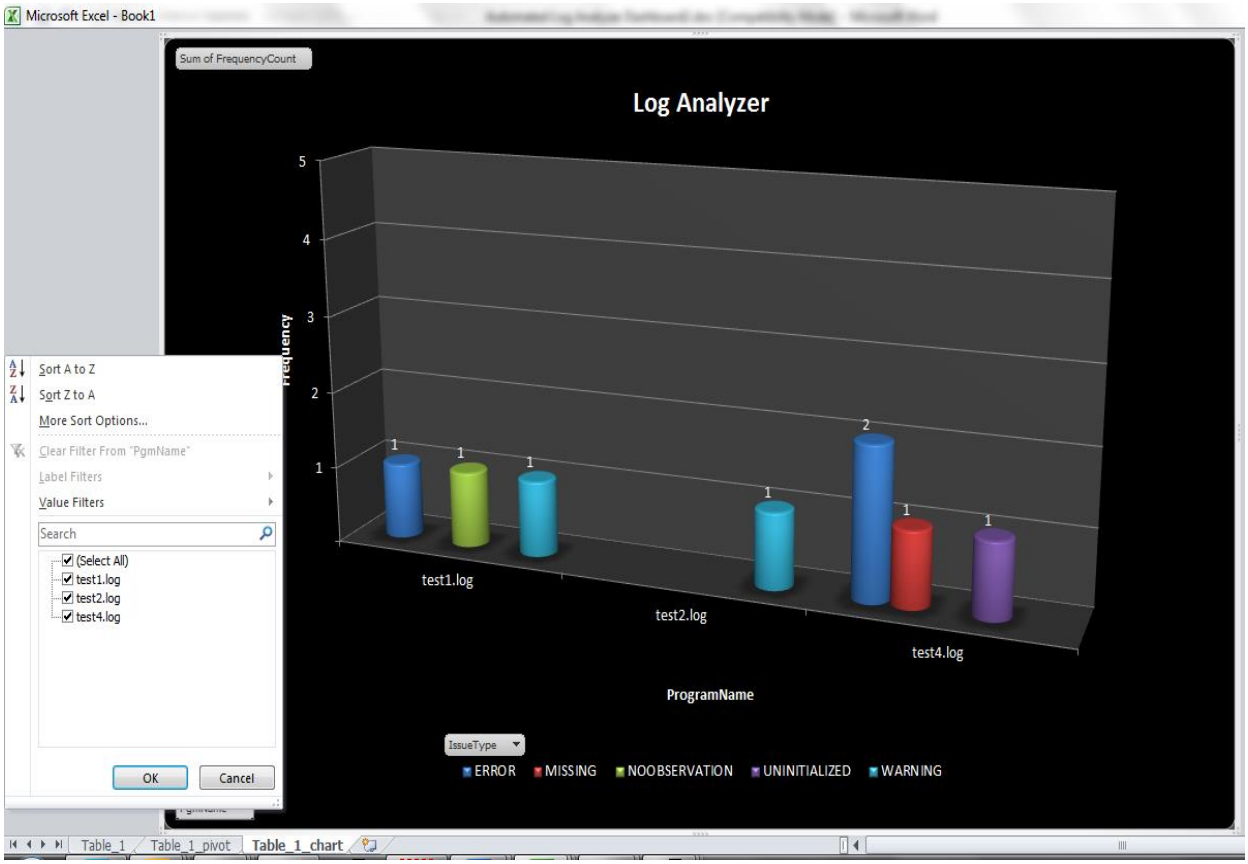
The second tab named Table_1_pivot creates a summary of all the system messages encountered during the execution of the SAS. For instance in the above report Column G Row 3 tells us how many issue messages did test1.log contain and Column B Row6 provides us information on how many Error messages did we encounter across all the programs.

TAB3 - TABLE_1_CHART: GRAPHICAL PRESENTATION

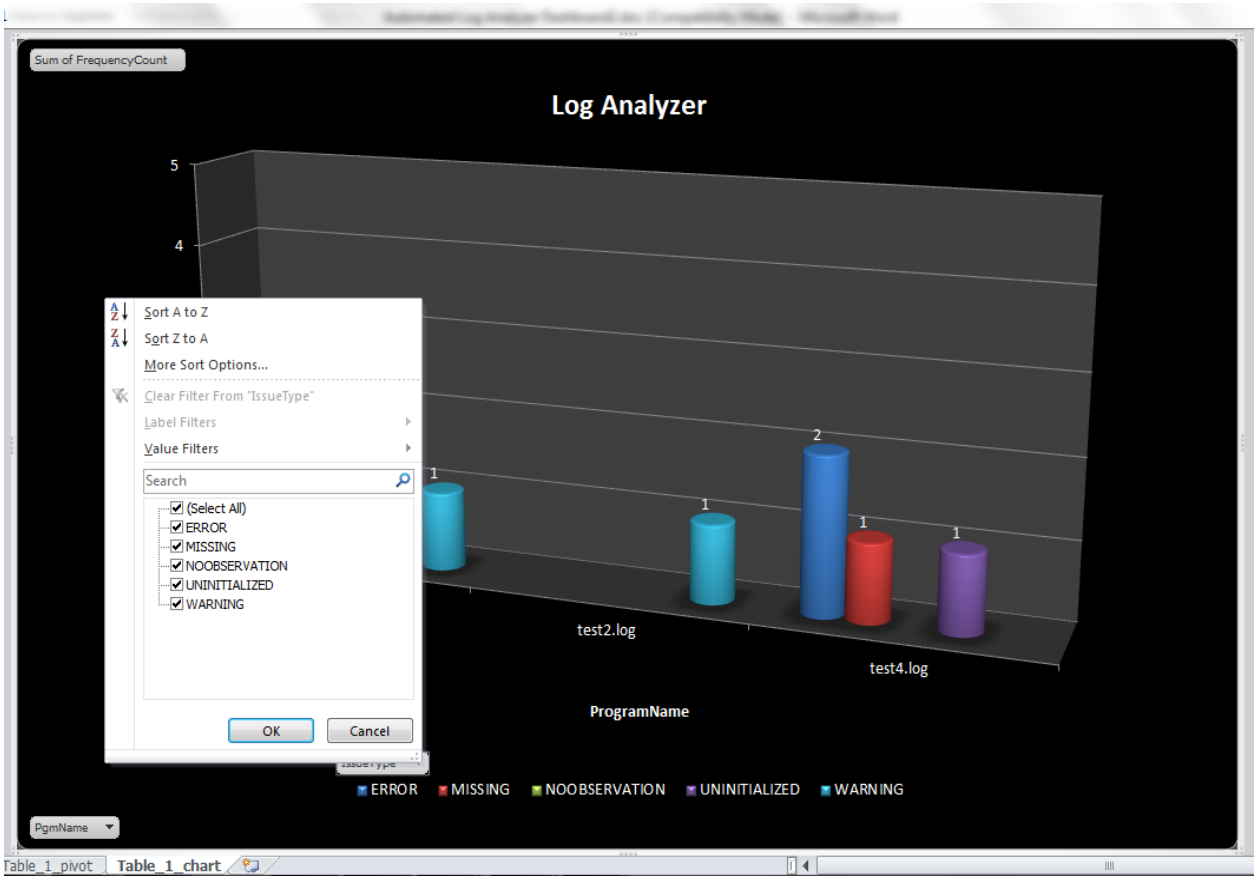


The last tab of the generated file is the graphical presentation on the number of issue messages encountered during the execution. The following section shows the various usages of the generated reports.

The last tab TAB3 - Table_1_chart of the file – graphical presentation is what is used for reviewing the issue messages generated during the execution. The tab provides the flexibility to view the number of issue message per program or across programs.



Display 1: Viewing issue message logs of each program



Display 2. : Viewing issue messages across logs

CONCLUSION

Automated Log Analyzer Dashboard provides an overall summary on various system generated messages in SAS logs through quality charts which will allow the programmer to quicken the validation process ensuring not only accuracy but also a significant reduction in the time taken to validate the programs.

CONTACT INFORMATION

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

Name: Palanisamy Mohan
Enterprise: Cognizant Technology Solutions
Address: Block F2, Manyata Embassy Business Park, Outer Ring Road,Rachenahalli
City, State ZIP: Bangalore, Karnataka, 560045
E-mail: palanisas@gmail.com

Name: Amarnath Vijayarangan
Enterprise: Genpact
Address: Block B Salarpuria Soft Zone Outer Ring Road, Bellandur
City, State ZIP: Bangalore, Karnataka, 560103
E-mail: 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

```
filename pth ".";

data LogList;

     length PgmName $ 50;

     dval=dopen('pth');

     if dval> 0 then do;

           count=dnum(dval);

           do i= 1 to count;

                PgmName=dread(dval,i);

                if upcase(scan(PgmName,2,'.'))='LOG' then output;

           end;

     end;

     keep PgmName;

run;

filename pth clear;

data LogAnalyzer(rename=(LogFile=PgmName));

     length IssueType $ 20;

     set LogList;

     LogFile1=PgmName;

     infile LogFile filevar=LogFile1 length=len end=done truncover;
```

```
      do while(not done);
        LogFile=LogFile1;
        FrequencyCount=1;
            input SystemMessage $varying2000. len ;
            IssueType='';
            SystemMessage=upcase(SystemMessage);
            if scan(SystemMessage,1,':')='ERROR' then IssueType='ERROR';
            if scan(SystemMessage,1,':')='WARNING' then IssueType='WARNING';
            if index(SystemMessage,'UNINITIALIZED') then IssueType='UNINITIALIZED';
            if index(SystemMessage,'MISSING VALUES WERE') 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,'EXTRANEOUS INFORMATION') then
                                                       IssueType='EXTRANEOUS';
            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 IssueType^='' then output;
      end;
      keep LogFile IssueType SystemMessage FrequencyCount;
run;

ods tagsets.tableeditor file="LogAnalyzer.html"
                        options(
```

```
                        button_text = "SAStoExcel"
                        pivotrow="PgmName"
                        pivotcol="IssueType"
                        pivotdata="FrequencyCount"
                        pivotdata_fmt="#,###"
                        pivotcharts="yes"
                        chart_type="cylindercolclustered"
                        chart_title="Log Analyzer"
                        chart_yaxes_title="Frequency"
                        chart_xaxes_title="ProgramName"
                        chart_legend="bottom"
                         chart_datalabels="value"
                         chart_style="42"
                         gridline="no"
                         chart_yaxes_maxscale='5'
                         chart_yaxes_minscale='0'
                         );
      proc print data=LogAnalyzer noobs;
            var PgmName SystemMessage IssueType FrequencyCount;
      run;
ods tagsets.tableeditor close;
```