

## Log Checks Made Easy

Yogesh Pande, Merck Sharp & Dohme Corp., Rahway, New Jersey

### **ABSTRACT**

It is Good Programming Practice (GPP) for a programmer to check the SAS® log for errors, warnings, and all other objectionable SAS NOTES. In order to successfully create tables, listings, and figures, the programmer must ensure that code is correct. The accuracy of the code, assuming that the program logic is correct, solely depends on SAS log. Using SAS macro language and Base SAS®, this paper will introduce a macro that will enable a programmer or statistician to check all SAS logs in a folder and tabulate the log issues by SAS log files using PROC REPORT.

*Keywords: SAS Log, Macro, report, infile, print*

### **1. MACRO PARAMETER EXPLANATION:**

This section of the paper explains in detail the key parameters macro macro '%mlogreport' to check all log files in a folder.

1. **Input file:** This parameter will give the location of the .log files that need to get checked. This key parameter cannot be kept as blank. Example: *Input\_file = C:\folder1\folder2\mylogfile.*
2. **Detail flag:** This parameter is type Y/N; default value is N. The detail flag parameter gives options whether a programmer wants a detailed report of the log checks or needs a simple report. If the parameter is Detail=Y, the log check output will have log issues and actual line numbers in the log file where the issue has been seen. If the parameter is Detail=N, it will have two columns; the first column will have a hyperlink of the log file, and the second column will indicate the issues seen in that log file. Please refer to section 4 (Various Displays Created by Macro) of this paper for the types of output created by macro '%mlogreport'.
3. **Output filename:** This parameter enables the programmer/statistician to give the desired file name to the output generated by the macro. The output will be in HTML format.
4. **Out file:** This parameter will enable the programmer/statistician to give the location where the user wants the output file to be seen. Example: *C:\folder1\folder2\mylogcheck.*
5. **Allog:** This parameter is type Y/N. The default value is N. This parameter is specially created within this macro to display all logs. The logs that have issues and logs that have no issues. This key parameter will give an idea about how many log files are present in the folder and how many log files have issues and how many log files have no issues. Please refer to section 4 (Various Displays Created by Macro) of this paper for the types of output created by macro '%mlogreport'.

### **2. TYPE OF PROBLEMATIC SAS LOG STATEMENT CHECKED BY MACRO:**

The following is a list of common SAS warnings, errors, and notes that are tracked by the macro '%mlogreport' thus enabling a programmer to easily identify which SAS program has issues and requires cleaning.

- 1) Errors
- 2) All types of Warnings
- 3) Uninitialized variables
- 4) Format not found
- 5) Merge statement has more than one repeat of by variables
- 6) Drop, rename list has never been referred
- 7) Decimal may have been shifted
- 8) Apparent symbolic reference not found
- 9) Overwritten variables
- 10) Use the length as a very first statement

### **3. EXAMPLE OF HOW TO CALL A MACRO:**

**Example 1:** `%mlogreport(input_file=%str(C:\folder1\folder2\logfiles)  
,detail_flag=n  
,output_filename=testing  
,out_file =%str(C:\folder1\folder2\logchecks)  
,allog = n);`

## Log Checks Made Easy, continued

### Explanation:

1. INPUT\_FILE in the macro call above checks the logs that are located at "C:\folder1\folder2\logfiles".
2. DETAIL\_FLAG has value n indicates that this macro will NOT generate line numbers where issues are found.
3. OUT\_FILENAME has value 'testing' indicates that after the mlogreport is executed the HTML file having all the results of the log checks will be stored in testing.html. (This macro only generates HTML output).
4. OUT\_FILE is giving location "C:\folder1\folder2\logchecks" in which testing.html will be stored.
5. ALLOG has value n, which indicates only log files having issues will be displayed in testing.html. (Please see section 4 for the screen shot)

**Example 2:** `%mlogreport (input_file=%str(C:\folder1\folder2\logfiles)  
,detail_flag=y  
,output_filename=testing  
,out_file =%str(C:\folder1\folder2\logchecks)  
,allog = y);`

### Explanation:

1. The only difference between the previous example and this example is DETAIL\_FLAG and ALLOG has value y.
2. The output testing.html will show line numbers for issues that are located in the log files and because ALLOG has value y it will also display those logs that have no issues. (Please see section 4 for the screen shot).

## 4. VARIOUS DISPLAYS CREATED BY MACRO:

Please see the screen shot below for the title that will be displayed for any of the outputs this macro generates.

**Number of Various Types of Issues found in the logs**  
**Total Number of Logs in the location given below: 6**  
[Log file location](#)

Explanation: The title generated by this macro gives all information about how many total log files the location folder has. The third title will ease the navigation for the user by clicking on the provided hyperlink; this hyperlink will directly take the user to the folder location in which all log files resides.

**Example 1:** `%mlogreport (input_file=%str(C:\folder1\folder2\logfiles)  
,detail_flag=n  
,output_filename=testing  
,out_file =%str(C:\folder1\folder2\logchecks)  
,allog = n  
);`

Below is a screen shot of the resulting output if the user calls %mlogreport macro.

Output	Name of the log file	Modified date	Log Message
Table	<a href="#">testlog1</a>	10Dec2014:08:35:52	WARNING: MCRLIB.TEMPLAT is not a template store! It will be ignored.
	<a href="#">testlog2</a>	10Dec2014:08:36:08	WARNING: MCRLIB.TEMPLAT is not a template store! It will be ignored.
	<a href="#">testlog3</a>	10Dec2014:08:36:19	WARNING: MCRLIB.TEMPLAT is not a template store! It will be ignored.
	<a href="#">testlog4</a>	10Dec2014:08:35:52	ERROR: SPEC Label Consistency for: BOREFR
			ERROR: SPEC Label Consistency for: CA125EFR
			ERROR: SPEC Label Consistency for: CA125RES
			ERROR: SPEC Label Consistency for: CA_SET
			ERROR: SPEC Label Consistency for: EORRBCAM
			ERROR: SPEC Label Consistency for: EORRBRM
			ERROR: SPEC Label Consistency for: ORR_SET
			ERROR: SPEC Label Consistency for: SUBTHERF

Figure 1. Screen shot of the output for macro call as per example 1 in section 3 above

## Log Checks Made Easy, continued

Explanation:

- 1) The first column in the output will give details about the output type this program is generating. Please note that the output will always be sorted based on this column. The sort order for the log files will be analysis datasets, figures, tables, listing, and other miscellaneous files.
- 2) The second column in the output of Figure 1 indicates the name of the SAS log file. Please note that this is a hyperlink. If there is more than one log file experiencing issues, each log file displayed will have its own hyperlink.
- 3) The third column will show the last modified date and time of SAS Log. This will help identify the user whether he/she is checking the most recent log file.
- 4) The fourth column gives the types of log issues associated with the log file. If a log file has errors, the row will be highlighted in red indicating that these logs need immediate review.

(Example 2 screen shot is continued below)

```
Example 2: %mlogreport(input_file=%str(C:\folder1\folder2\logfiles)
, detail_flag=y
, output_filename=testing
, out_file =%str(C:\folder1\folder2\logchecks)
, allog = y);
```

Below is a screen shot of the resulting output if the user %mlogreport macro.

Output	Name of the log file	Modified date	Line number where the log issue is seen	Log Message
Table	<a href="#">testlog1</a>	10Dec2014:08:35:52	61	<b>WARNING: MCRLIB.TEMPLAT is not a template store! It will be ignored.</b>
			17965	<b>ERROR: SPEC Label Consistency for: BOREFR</b>
			17966	<b>ERROR: SPEC Label Consistency for: CA125EFR</b>
			17967	<b>ERROR: SPEC Label Consistency for: CA125RES</b>
			17968	<b>ERROR: SPEC Label Consistency for: CA_SET</b>
			17969	<b>ERROR: SPEC Label Consistency for: EORRBCAM</b>
			17970	<b>ERROR: SPEC Label Consistency for: EORRGBRM</b>
			17971	<b>ERROR: SPEC Label Consistency for: ORR_SET</b>
			17972	<b>ERROR: SPEC Label Consistency for: SUBTHERF</b>
Table	<a href="#">testlog2</a>	10Dec2014:08:36:08	61	<b>WARNING: MCRLIB.TEMPLAT is not a template store! It will be ignored.</b>
Table	<a href="#">testlog3</a>	10Dec2014:08:36:19	61	<b>WARNING: MCRLIB.TEMPLAT is not a template store! It will be ignored.</b>
Table	<a href="#">testlog4</a>	10Oct2014:03:43:26	.	<b>This log file is clear of Errors, Warnings and Other Objectionable Notes</b>
Table	<a href="#">testlog5</a>	21Nov2014:12:01:29	.	<b>This log file is clear of Errors, Warnings and Other Objectionable Notes</b>
Table	<a href="#">testlog6</a>	24Nov2014:01:37:26	.	<b>This log file is clear of Errors, Warnings and Other Objectionable Notes</b>

Figure 2. Screen shot of the output for macro call as per example 2 in section 3 above.

Explanation:

- 1) For the macro call as per Example 2, first, second, third and fifth column remains similar to what is seen in example 1, the only difference is because detail\_flag = Y, it has generated additional column that will indicate the line number within the log file which has the issue as described in the column named log message.
- 2) Another difference between Example 1 and this example is, this HTML output is displaying all logs, even though some of the logs have no issues. If the log has no issues, you can notice that line number column is blank and the text string in Log message indicates "This log file is clear of Errors, Warnings and other Objectionable Notes".

## **Log Checks Made Easy, continued**

As shown above in Figure 1 and Figure 2 the output format generated by macro %*mlogreport* provides the programmer with the hyperlink of the individual log file having issues which requires cleaning. The most important advantage of having a hyperlink of the individual log files is the navigation becomes very easy. Thus the time need for a programmer to go back to the folder and look for that particular log file having issues has been minimized.

### **5. MACRO CODE:**

Please click on the below link for the partial SAS code. I will be happy, if interested, to discuss the logic behind my program for enabling full use of this technique on checking SAS log files.

<https://sites.google.com/site/sasfilesyogesh/home/pharmasug-2016>

### **6. CONCLUSION:**

This paper has explained how a macro can check various log issues such as errors, warnings, and all other objectionable SAS log NOTES. Further, this paper explains ways to check multiple log files by using the SAS macro tool. This paper also demonstrated the ability to create an organized output of the checks used for accessing the log files in which the log issues appear. It also gives an idea to SAS programmers the processes to be carried out in a log file while writing a SAS code for generating tables, listing, and figures. The goal of this paper is to create a mechanism for checking multiple logs in a folder of a project.

### **7. REFERENCE:**

1. SAS/MACRO Software: version 9.1.3, SAS® Institute Inc., Cary NC
2. SAS/BASE Software: Version 9, SAS® Institute Inc., Cary NC
3. SAS® Guide to Report Writing: Examples, Second Edition
4. SAS 9.1 Macro Language Reference
5. SAS Programming in Pharmaceutical industry

### **CONTACT INFORMATION:**

Your comment and questions are valued and encouraged.

Yogesh Pande

Sr. Scientist, Statistical Programming

Merck Sharp & Dohme Corp.

E-mail: [yogesh.pande@merck.com](mailto:yogesh.pande@merck.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.