

Manage SAS Log in Clinical Programming

Yuping Wu, PRA Health Science

ABSTRACT

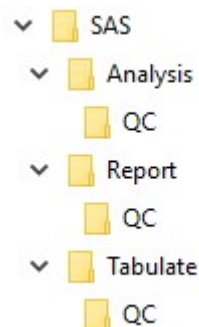
This paper introduces a method to manage SAS log in clinical programming. It is conducted by three macros. The first macro is placed at the beginning of the program. It automatically detects the type of the program that is being run and specifies the path to store the logs based on the program type. The second macro is placed at the end of the program. It prints the logs and lists the log issues in the current log window. Therefore, programmers can know the log status instantly when they are developing or running the individual program. The third macro is used to check log issues in batch. It scans stored log files of the selected processes such as SDTM or ADaM or Table, Figure and Listing (TFL) or all of them, and produces a status report that highlights the programs with log issues. This method presents an automatic way to manage SAS log and allows programmers to easily detect and fix log issues in current log window and identify programs with log issues in batch report.

INTRODUCTION

Clinical programming consists of three major processes, SDTM, ADaM and TFL. A typical clinical study may involve in creating 25-35 SDTM, 15-30 ADaM data sets and 100-300 TFLs and developing 200-300 production and validation programs. Each program may need to be updated for data refreshing, programming issues and updated specification or annotation and to be rerun for multiple dryruns and interim analysis. It can be a challenging and time-consuming work to ensure clean log for that many programs in multiple deliveries. A number of papers have discussed how to detect potential log issues based on the consistent pattern of SAS log [1] [2]. Most of organizations also developed macros to detect and summarize log issues. A common method used for log management is to specify log path in initiation program and program name in the program that is being developed, print the logs to specified folders and then use log checking macro to analyze logs and generate status report. There are a number of drawbacks with this method. For example, wrong program name is specified due to human error, especially for validation programs, 'qc_' is often missed in the programs, proc printto is commented out which will prohibit refreshing the log, Log is not available in current log window you will have to open the log in a different window to visually examine if there is log issue. It is tedious and time consuming to check large number of programs for these human errors and visually examine log issue for each individual program. This paper presents an automatic method for log management. It consists of three macros. The first macro is used to specify the log path and the second macro detects log issues and summarizes them in the current log window. There is no macro parameter for these two macros. They are done in automatic fashion. The third macro allows programmers to check and summarize log issues in batch. With this method programmers don't need to specify log path and program name and they can easily detect log issues in the current log window, which greatly reduces human error and improves efficiency.

PREREQUISITE

This log management method requires that the SAS programs are stored per category. ADaM programs are stored in a different folder from SDTM programs and all TFL programs are store in the same folder. The folder structure is set up in the same way for all studies. Therefore, it can be easily applied to all studies. The display 1 shows the folder structure used for programs at PRA Health Science. The analysis folder stores ADaM programs, the tabulate folder stores SDTM programs and the report folder stores TFL and narratives programs. The display 2 shows the folder structure for data sets, outputs, logs and proc compares. The logs



Display 1. Folder Structure for SAS Programs

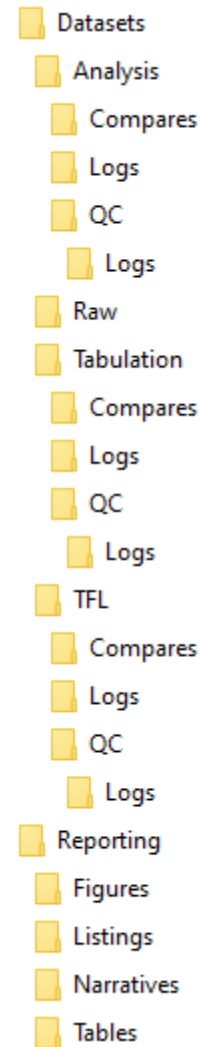
folder under Analysis is used for storing the logs of primary ADaM programs. The Logs folder under QC under Analysis stores the logs of validation ADaM programs. Macros developed in this paper retrieve the program path and use the path information to determine the type of program and specify the path for log and proc compare.

THE FIRST MACRO

This method contains three macros. They are run at different stage of program and sever for different purpose. The macros in this method are developed in SAS Enterprise Guide (EG). The first macro uses SAS system macro variable, `_SASPROGRAMFILE` to retrieve the full path and name of the SAS program that is currently being run. For batch running all SDTM, ADaM and TFL programs interactively, the macro uses `SASHELP.VEXTFL` file to retrieve the path and name of the current program. `VEXTFL` contains information about the last file opened by the SAS system. If you are running all SDTM, ADaM and TFL program in a macro the last file will be the SAS program that is be running. In the windows operating environment, you may use `%sysget(sas_execfilename)` or `%sysfunc(getoption(sysin))` to retrieve the path and name of the program that is being run. The macro then uses the path information to determine the type of the program. For example, if it is from the Analysis folder it will be considered as primary ADaM program, if it is from Analysis QC folder it will be considered as validation ADaM program. The macro then specifies the log path and proc compare path based on the type of the program. The macro also retrieves the program name and uses it for the name of the log output. For other type of programs than SDTM, ADaM and TFL log will be stored in a temporary folder, `SASHELP.VSLIB`, which will be removed every 24 hours. Compared to traditional method this macro uses the information of the consistent folder structure for storing the programs to automatically determine the program type and derive the log path and name. It greatly reduces human error such as forget to specify the program name, specify a wrong name, comment out proc printto procedure. This macro does not have macro parameters. It can be easily used in any type of statistical programming. This macro should be placed at the beginning of the program and used together with the second macro.

THE SECOND MACRO

This macro should be placed at the end of the program. It retrieves the log from the specified folder and prints the log to the current log window. Log is then scanned for any log issues. Here are the rules we use to determine if it is issue. Log text starts with `ERROR` or `WARNING` or contains any one of the following key words, `MISSING VALUES`, `UNINITIALIZED`, `INVALID`, `ARGUMENT`, `WITH REPEAT OF BY VALUES`, `VALUES HAVE BEEN CONVERTED`, `W.D. FORMAT`, `FORMATTING EXCEPTIONS OCCURRED`, `WAS NOT FOUND OR COULD NOT BE LOADED`, `INFOR: THE VARIABLE`, `WILL BE OVERWRITTEN`, `TRUNCATED`, `MATHMEMATICAL OPERATIONS COUNLD NOT`, `DIVISION BY ZERO`, `NOTE: MERGE`, `ODERING BY AN ITEM THAT DOSEN'T APPEAR IN`, `OUTSIDE THE AXIS RANGE`, `RETURNING PREMATURELY`, `UNKNOWN MONTH FOR`, `QUERY DATA`, `??`, `QUESTIONABLE`. If log issues are detected the line numbers where the log issues are located and the first line of log text will be printed to the end of the current log window. If there is no log issue the text 'CLEAN LOG' will be printed. The display 3 shows an example of log issue list at the end of the current



Display 2. Folder Structure for Data Sets, Log and Compare

log window. This macro allows programmers to detect if there is an issue, where are the issues and what are the issues and quickly fix the log issues when they are working on or running the programs.

```
=====Log Summary=====
---LINE# 542 NOTE: MERGE STATEMENT HAS MORE THAN ONE DATA SET WITH REPEATS OF BY VALUES.
---LINE# 551 NOTE: MERGE STATEMENT HAS MORE THAN ONE DATA SET WITH REPEATS OF BY VALUES.
---LINE# 687 NOTE: MISSING VALUES WERE GENERATED AS A RESULT OF PERFORMING AN OPERATION ON MISSING VALUES.
---LINE# 758 NOTE: VARIABLE MISSING IS UNINITIALIZED.
=====End of Log Summary=====
```

Display 3. The Log Summary in the Current Log Window.

THE THIRD MACRO

The first two macros are always used together in the same program. The third macro is used to check log issues in batch, which is independent of the first two macros. This macro will read into SAS the log files in different folders and examine them individually for the log issues using the same rule used in the second macro. Log issue checking by this macro is kept as simple as possible. It will not show how many log issues and what they are in the program but check if there is any log issue in the program. If log issue is found it will be marked as 'Y' and color highlighted in the status report. If no log issues are found it will be marked as 'N'. To resolve the issue you may run the individual program and the second macro will generate the detail report for the log issues in the current log window. This macro have one macro parameter that allows programmers to select which process they want to check for the log issue. They may choose all three processes, SDTM, ADaM and TFL or any individual process.

Program Name	Program Type	Primary		Validation	
		Primary Log Issues?	Date Log Created	Validation Log Issues?	Date Log Created?
AE	SDTM	Y	12/01/2020	N	12/01/2020
APDM	SDTM	N	12/01/2020	N	12/01/2020
DD	SDTM	Y	12/01/2020	Y	12/01/2020
ADDILI	ADAM	N	12/08/2020	N	12/08/2020
ADSL	ADAM	N	12/08/2020	Y	02/17/2021
T_AE	TFL	N	03/04/2021	N	03/04/2021
T_HIT6	TFL	Y	03/04/2021	N	03/04/2021
T_PFS	TFL	Y	03/04/2021	Y	03/04/2021

Display 4. The Log Status Report

HOW TO USE

The macro is very easy to use. There is no macro parameters for the first and second macro. There is one macro parameter for the third macro to allow you to select the processes to check log issues. You may select only SDTM or ADaM or TFL or all three processes. It is also easy to copy to different studies since the folder structure is usually the same for all studies within an organization.

CONCLUSIONS

Organizations typically use the same folder structure to store statistical programs, logs, proc compares and outputs for their studies. We use this information to develop automatic method for log management. The macros used in this method retrieve program name, determine program type, specify folders for log deposition, summarize log issues in current log window and generate batch log report. All of these are done in automatic fashion which greatly reduces human error and improves efficiency.

REFERENCES

[1] Suzanne Humphreys. "%LOGCHECK: a Convenient Tool for Checking Multiple Log Files. PharmaSug, 2008.

[2] Sridhar R Dodlapati, Kiran Kumar Karidi, Mahipal R Vanam. "LOG CHECKING: What to check and why?", NESUG, 2010.

CONTACT INFORMATION

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

Yuping Wu
PRA Health Science
9755 Ridge Drive
Lenexa, KS 66219, USA
wuyuping@prahs.com

Any brand and product names are trademarks of their respective companies.