

## A Diagnostic Report for SAS Submission Programs

Yang Gao, Christine Teng, Merck & Co., Inc., Rahway, NJ, USA

### ABSTRACT

To submit a data package to a health authority for regulatory approval, sponsors must ensure analysis programs meet regulatory compliance requirements and good business practices. Each company has their own standard operating procedure (SOP) as well as checking tools that assist in the development of regulatory-compliant SAS programs. The checking tools are used to ensure these SAS programs are ready for internal and external audits.

This paper introduces a SAS utility macro with multiple features that provides detailed information for each SAS program. The key features include: (1) checking correspondence between SAS programs and associated log, lst, programs (in submission ready format) as well as analysis outputs, (2) retrieving a SAS program's last modified date (system date), program version date and last revision date in development and production area to identify inconsistency, (3) producing a summary of output counts (SAS log, lst, etc.) for each SAS program. The outputs from this utility macro will be summarized in an Excel workbook using SAS ODS. This utility macro facilitates the process of identifying programming compliance concerns. It substantially reduces the resources and time needed for reviewing potential issues for submission programs during the analysis and reporting lifecycle. This diagnostic Excel workbook provides the compliance status of SAS programs in an organized format.

### INTRODUCTION

A large-scale clinical trial may have many SAS programs. It is mandatory that all SAS programs meet the regulatory compliance requirements. Ensuring the regulatory compliance of all programs is a complex and challenging task. Some companies may have checking tools to ensure all SAS programs used to produce analytical reports are compliant and are ready for audits.

When approaching a database lock, clinical programmers run a series of compliance checks for constructed SAS code in study folders. The checking tool may have some limitations in terms of reporting specific compliance findings for each program. For instance, the reported findings do not provide descriptions in detail; programmers need to go to another Excel spreadsheet to refer to the issue details and possible solutions; the findings are not listed by categories in an organized way. This SAS utility macro provides supplemental features for existing checking tools.

Checking and addressing the compliance concerns can be time consuming. Programmers have to navigate to different subfolders, open programs, identify the compliance findings and make updates to resolve these compliance issues. This process may not be an effective way to ensure quality and efficiency. Instead, it is highly desirable to create a SAS utility macro that can facilitate the process of checking and addressing compliance findings. The compliance considerations include checking the existence of SAS programs with corresponding log files in a specific folder, confirming SAS programs running in a logical order in development and validation areas, and ensuring consistent file date stamps between test and production area. The number of output files between development and validation also needs to be verified for consistency.

### PROPOSED SOLUTION

This paper introduces a macro that facilitates the automation of these high-volume tasks and boosts accuracy and efficiency of the working process. With the aid of this utility macro, the resources, efforts and time spent on compliance checking for submission programs are substantially reduced.

This utility macro has many features:

- Checks correspondence between SAS programs and associated outputs within a directory
- Lists program’s system dates, and version as well as revision dates in development and production folders
- Lists counts of log, lst, output files for each development program and corresponding validation program
- Creates an Excel file to serve as a reporting tool for managing regulatory compliance of selected SAS programs as described above.

## OUTPUT 1: CORRESPONDENCE BETWEEN SAS PROGRAMS AND ASSOCIATED LOG, LST, PROGRAMS IN SUBMISSION READY FORMAT AS WELL AS ANALYSIS OUTPUTS

This feature captures paths and names for all files with different extensions, including the SAS programs, SAS datasets, rtf files, log files, lst files and programs in submission format (Gen program). Listing all SAS programs, and associated log, lst, rtf files together enables the checking of all required files for the package in one central location. Since the output is organized in one central file, we can spot the missing components right away. The output illustration is depicted as below in Figure 1.

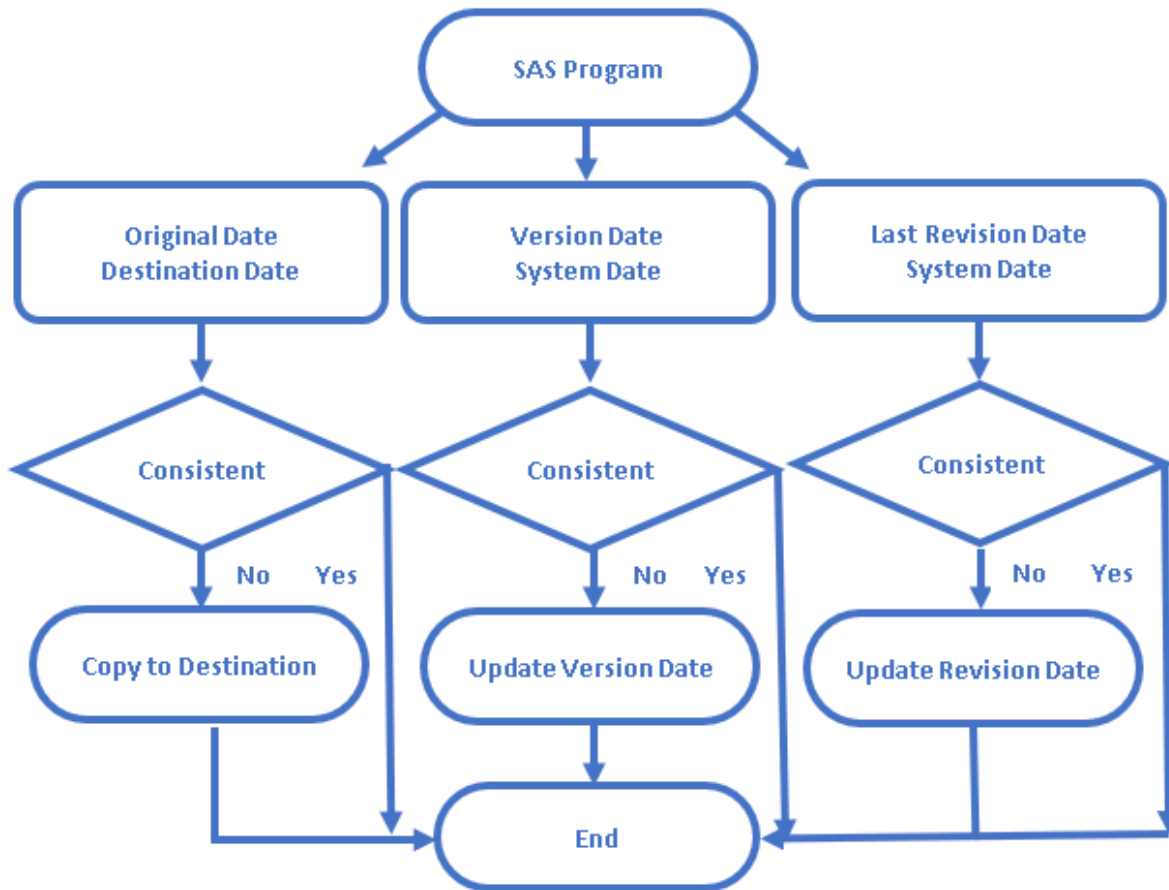
Program Location	Output Location	Log Location	Lst Location	Gen Location
pgmsetup/adae.sas	dataanalysis/adae.sas7bdat	outlog/adae.log	outlist/adae.lst	outlist/adae0gen.sas
pgmsetup/addili.sas	dataanalysis/addili.sas7bdat	outlog/addili.log	outlist/addili.lst	outlist/addili0gen.sas
pgmsetup/adex.sas	dataanalysis/adex.sas7bdat	outlog/adex.log	outlist/adex.lst	outlist/adex0gen.sas
pgmsetup/adexsum.sas	dataanalysis/adexsum.sas7bdat	outlog/adexsum.log	outlist/adexsum.lst	outlist/adexsum0gen.sas
pgmsetup/adpm.sas	dataanalysis/adpm.sas7bdat	outlog/adpm.log	outlist/adpm.lst	outlist/adpm0gen.sas
pgmsetup/adsl.sas	dataanalysis/adsl.sas7bdat	outlog/adsl.log	outlist/adsl.lst	outlist/adsl0gen.sas
pgmanal/call0apr0ae0summary.sas	outtable/s0apr0ae0summary.rtf	outlog/s0apr0ae0summary.log	outlist/s0apr0ae0summary.lst	outlist/s0apr0ae0summary0gen.sas
pgmanal/call0apr0disp0nats.sas	outtable/d0apr0disp0nats.rtf	outlog/d0apr0disp0nats.log	outlist/d0apr0disp0nats.lst	outlist/d0apr0disp0nats0gen.sas
pgmanal/call0apr0ds.sas	outtable/d0apr0ds.rtf	outlog/d0apr0ds.log	outlist/d0apr0ds.lst	outlist/d0apr0ds0gen.sas
pgmanal/call0apr0rainfall.sas	outtable/s0ae0rainfall0all.rtf	outlog/s0ae0rainfall0all.log	outlist/s0ae0rainfall0all.lst	outlist/s0ae0rainfall0all0gen.sas

**Figure 1. Retrieved SAS Program and Associated Outputs (Rtf, Log, Lst and Gen programs) to Check the Correspondence of SAS Programs and Associated Outputs**

## OUTPUT 2: LISTINGS OF PROGRAM’S LAST MODIFIED DATE (SYSTEM DATE) IN DEVELOPMENT AND PRODUCTION FOLDERS, VERSION DATE AND LAST REVISION DATE

A common and important compliance finding is that SAS programs do not have consistent dates. In particular, the different scenarios for consistency in dates include:

- A SAS program copied from the original folder to a destination folder must have the same system date.
- The SAS program date stamp is consistent with the version dates in the program header.
- The SAS program date stamp is consistent with the last revision date in the program header.



**Figure 2. Flow Chart to Check and Address Inconsistent System Date and Version as well as Last Revision Date**

To address these types of compliance concerns, a graphical representation of steps taken are shown in Figure 2. In the first scenario, the development program passing validation does not have the same file date stamp when copied to a destination folder. To address this identified concern, programmers must copy the most recent version of SAS programs to the destination folder. In the second scenario, the version date in the program header and the file date stamp are not consistent. The inconsistency indicates that the program was updated, but the version date was not updated in the program. In the third scenario, the SAS program revision date is different than its corresponding system date. All affected programs with inconsistencies need to be updated and reran.

In order to track inconsistent dates for all SAS programs in a project, this SAS utility macro can be used to address all three cases mentioned above. This macro retrieves the date stamps using FINFO function. The version date and revision date information can be extracted using FILENAME statements to search specific strings in the header. The code segments using FINFO function and FILENAME statement are shown below:

```

%macro filelmd(dir=);
  %let rc=%sysfunc(filename(filrf,&dir));
  %let did=%sysfunc(dopen(&filrf));

```

```

%do i=1 %to %sysfunc(dnum(&did));
%let name=%qsysfunc(dread(&did,&i));
%let rc1=%sysfunc(filename(onefile,"&dir/&name"));
%let fid=%sysfunc(fopen(&onefile));

%if &fid ne 0 %then
%do;
%let ModifDT1=%sysfunc(finfo(&fid,Last Modified));
%let fidc1=%sysfunc(fclose(&fid));
%let rc1=%sysfunc(filename(onefile));
%end;
%end;
%mend file1md;

data textfrompgm;
length filepath _filepath $200.;
infile sfile filename=_filepath lrecl=200 end=done;
do while (not done);
input rec $200.;
filepath=strip(_filepath);
output;
end;
run;

```

Each SAS program's date stamp, version and revision dates are listed within one central file. This central file is the single source that programmers need to check for all date inconsistencies. Figure 3 below shows how one can easily target the date inconsistency findings for adae.sas as an example. One of the compliance findings is that the development and production folders have different versions of the SAS program. Additionally, the version date is not consistent with the system date.

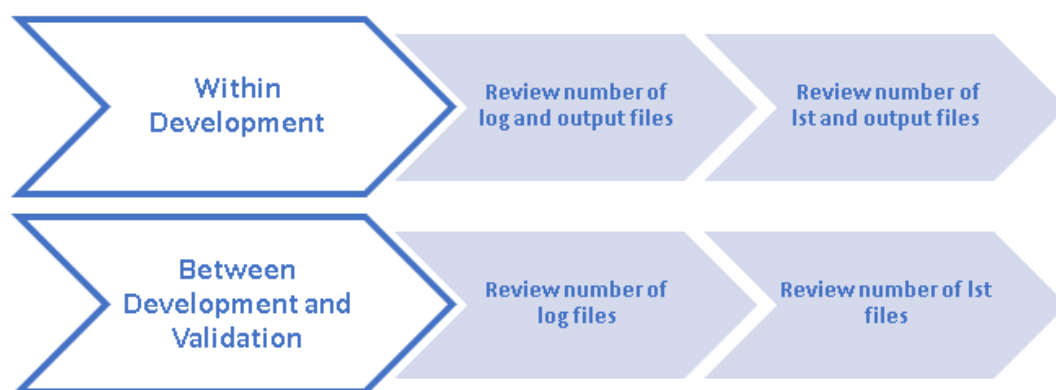
To further facilitate the work on addressing the date stamp inconsistency findings, hyperlinks are available for quick and easy navigation to each SAS program. The details on how to establish the hyperlinks for files are described in "A Tool to Organize SAS Programs, Output and More for a Clinical Study". by Y. Gao.

Prod Program Location	Dev Last Modified	Prod Last Modified	Dev Version Date	Dev Revision Date
pgmsetup/adae.sas	2021-01-05	2020-12-01	2020-06-19	.
pgmsetup/addili.sas	2020-11-20	2020-11-20	2020-11-20	2020-11-20
pgmsetup/adex.sas	2020-07-23	2020-07-23	2020-07-23	2020-07-23
pgmsetup/adexsum.sas	2020-10-02	2020-10-02	2020-08-28	2020-08-28
pgmsetup/adpm.sas	2020-12-07	2020-12-07	2020-07-23	2020-07-23
pgmsetup/adsl.sas	2020-12-16	2020-12-10	2020-07-23	2020-12-10
pgmanal/call0apr0ae0summary.sas	2020-11-18	2020-11-18	2020-11-18	2020-11-18
pgmanal/call0apr0disp0nats.sas	2020-11-18	2020-11-18	2020-11-18	2020-11-18
pgmanal/call0apr0ds.sas	2020-11-18	2020-11-18	2020-11-18	2020-11-18
pgmanal/call0apr0rainfall.sas	2020-07-24	2020-07-24	2020-07-24	2020-07-24

**Figure 3. A List of System Dates and Documented Dates for SAS Programs**

## OUTPUT 3: COUNTS OF LOG, LST, OUTPUTS FOR EACH DEVELOPMENT SAS PROGRAM AND CORRESPONDING VALIDATION PROGRAM

Another frequently reported compliance concern is that log files are missing in a specific folder or the number of log files for same macro are different between development and validation. This utility obtains all files with different extensions in specific folders, the number of logs as well as the number of rtf outputs can be counted by individual SAS programs. If the numbers for a specific SAS macro are not consistent, the file location is available to route to that specific folder to investigate the inconsistencies. Similarly, the number of lst files can be compared to the rtf outputs. We expect one log and one lst for each one corresponding rtf output. If any discrepancies are identified, they can be checked, tracked and updated accordingly.



**Figure 4. Workflow to Check the Number Consistency of Output, Log and Lst Files**

Once the identified finding is addressed on development side, one must check the consistency of the numbers between development and validation. Some discrepancies in the number of log or lst files have been seen previously. The number of log files from two sides can be obtained by using this utility macro. The step by step process is shown in Figure 4. Like other steps, the number of rtf files and log files can be listed by adding additional columns in the central file shown in Figure 5.

File in Development	File in Validation	Log Counts (Dev/Val)	Lst Counts (Dev/Val)	Output Counts (Dev/Val)
pgm/adae.sas	pgm/dbl0adae.sas	1/1	1/1	
pgm/addili.sas	pgm/dbl0addili.sas	1/1	1/1	
pgm/adex.sas	pgm/dbl0adex.sas	1/1	1/1	
pgm/adexsum.sas	pgm/dbl0adexsum.sas	1/1	1/1	
pgm/adpm.sas	pgm/dbl0adpm.sas	1/1	1/1	
pgm/adsl.sas	pgm/dbl0adsl.sas	1/1	1/1	
pgm/call0apr0ae0summary.sas	pgm/dbl0apr0ae0summary.sas	1/2	1/2	1
pgm/call0apr0disp0nats.sas	pgm/dbl0apr0disp0nats.sas	1/1	1/1	1
pgm/call0apr0ds.sas	pgm/dbl0apr0ds.sas	1/1	1/1	1
pgm/call0apr0rainfall.sas	pgm/dbl0apr0rainfall.sas	1/1	1/1	1

**Figure 5. A List of Counts of Output Files for SAS Programs**

## CONCLUSION

It is very common to have multiple programmers supporting a submission study, especially in a large-scale study. It is ideal to have a tool that enables us to check for compliance and quickly identify the

source of concerns. This diagnostic tool is instrumental in managing the life cycle of program development. Compliance checks should be done regularly to ensure that programmers have good habits in following compliance expectations. This paper presents a powerful macro which can minimize manual efforts in identifying details of compliance concerns for SAS programs with their corresponding output files and versions. This utility macro substantially enhances the accuracy, reliability and efficiency of checking and addressing compliance concerns for a clinical study.

## REFERENCES

Gao, Y. 2020. "A Tool to Organize SAS Programs, Output and More for a Clinical Study". PharmaSUG.

Shetty, S. 2014. "Macro to check Audit compliance and standards of SAS® programs". PharmaSUG.

Lal, R. 2011. "Project Automation and Tracking Using SAS". PharmaSUG.

Qi, LJ and Donthi, B. 2019. "Automating SAS Program Table of Contents for Your FDA Submission Package". PharmaSUG.

Reap, C. 2005. A Regulatory Compliant Process for Developing SAS-Based Reports, Proceedings of the 30th Annual SAS Users Group International (SUGI) 2005 Conference.

List All Files within a Directory Including Subdirectories.

<https://documentation.sas.com/?docsetId=mcrolref&docsetTarget=n0js70lrkxo6uvn1fl4a5aafnlgt.htm&docsetVersion=9.4&locale=en>

Retrieve the File Size, Create Time, and Last Modified Date of an External File.

<https://documentation.sas.com/?docsetId=mcrolref&docsetTarget=n1wyzhuyjmwfsun1iwqnluezhq9.htm&docsetVersion=9.4&locale=en>

## CONTACT INFORMATION

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

Name: Yang Gao

Enterprise: Merck & Co., Inc. Address: 126 E. Lincoln Avenue

City, State ZIP: Rahway, NJ 07065-4607 USA

Work Phone: 732-594-8120

E-mail: [yang.gao5@merck.com](mailto:yang.gao5@merck.com)

Web: [www.merck.com](http://www.merck.com)

Name: Christine Teng

Enterprise: Merck & Co., Inc. Address: 126 E. Lincoln Avenue

City, State ZIP: Rahway, NJ 07065-4607 USA

Work Phone: 732-594-4425

E-mail: [christine\\_teng@merck.com](mailto:christine_teng@merck.com)

Web: [www.merck.com](http://www.merck.com)

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

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.