A SAS Macro for Checking the Status of Table, Figure and Listing (TFL) Programming

Yuping Wu, and Sayeed Nadim, PRA Health Science

ABSTRACT

A clinical study report (CSR) typically involves in generation of hundreds of TFLs. To efficiently manage such large batch of outputs, Lead programmers and managers often need to know the current status of each programs and outputs. Many organizations may use an Excel based tracking document where production and validation programmers can enter the TFL status. However, for a CSR delivery that usually takes several months to prepare, such document is often out of status due to data addition, ADaM updating, miscommunication between production and validation programmers, etc.. This paper introduces a tool that can efficiently monitor the current status of the TFL programs for their logs, proc compare reports, program and output creation.

INTRODUCTION

The CSR is a crucial document in the drug development and regulatory submission process. For most of clinical trials a CSR typically consists of hundreds of TFLs and takes several years to complete. During the course of a clinical trial there are multiple deliveries of CSR and each delivery may require different TFLs and involve rerunning old programs and developing new programs. Knowing the current status of each programs and outputs before programing starts will help lead programmer to manage resource needs and ensure on time delivery. An Excel based tracking document is widely used and it is also very convenient because primary and validation programmer can directly enter TFL status. For a new study and the data cutoff date is a past date this document can work fine. However, if it is a new study that take a long time to finish this document is often out of date by the time of delivery due to the addition of clinical data, AdaM updating and macro updating etc.. For a repeated delivery of an old study lead programmers often want to know the current status of programs before programming starts, therefore they can estimate how many resources they will need to implement the updating for the current delivery. This paper introduces a tool to quickly examine the program/output status for the current delivery, including checking if programs and output exist, primary and validation logs are clean and proc compare status. It will enable lead programmer to have a picture of current program status, for example, how many new programs are needed, how many outputs are not matched and how many programs have log issues which will help manager and lead programmer to determine the resource needs. This tool can also help lead programmer to double check the final status of all programs and outputs before delivery and to monitor the programming progress.

PREREQUISITE

The TFL macro requires a file that store a complete list of TFLs and their output and program names. The file can be the one used to store titles and footnotes or Table of Contents (TOC) used for programmer to enter the status and comments both of which are common files used in many organizations. We used TOC file here and a flag can be added to allow lead programmers to easily select TFLs for each different deliveries. The complete list of TFLs from TOC file will be used in the status report. The TFL macro is only used for monitor the status of TFLs. All SDTMs and ADaMs are assumed to be up to date.

CONTENTS OF THE TFL MACRO

We developed this macro to monitor the status of an old and large study that requires frequent data refreshing and submission. Before each delivery we will run this macro to get the status report for all programs and outputs, therefore we know how many new programs are needed, how many programs are not matched and how many of them have log issues. These question will help us to quickly figure out how

many resources we need to work on this delivery. We have also use it to confirm the program status before the delivery.

Figure 1 shows the process of the TFL macro. It consist of three parts, batch run TFL programs, check for issues in programs and outputs and generate status report.

BATCH RUN TFL PROGRAMS

To check the current status of the programs and outputs, we batch run all existing TFL programs to refresh all outputs, logs and proc compare reports. The TFL macro will identify sas programs in the folders for storing both primary and validation TFL programs and run all of them and have their logs and proc compare reports to be deposited in the specified locations. In our case, all primary TFL programs are stored in the same folder while all validation TFL programs are stored in the another same folder. If table, figure and listing programs are in different folders the same procedure should be repeated for different folders. Although we run all existing TFL programs the status report will only include TFLs selected from TOC file. Here is part of codes in the TFL macro for batch run.

```
***batch run primary programs;
x "cd &report";
filename sasprog pipe 'dir /B';
data sasprog;
  length progname $200.;
  infile sasprog LRECL=200 truncover;
  input progname;
  if index(upcase(progname), ".SAS") and (upcase(progname)=:'L_' or
  upcase(progname)=:'F_' or upcase(progname)=:'T_') then do;
    progname=scan(progname, 1, '.');
    output;
  end;
run;

proc sort data=sasprog;    ***all existing primary tfl programs;
  by progname;
run;

data _null_;
  set sasprog end=last;
  call symputx('prog'||strip(put(_n_, best.)), progname);
  if last then call symputx('np', _n_);
run;

%do j=1 %to &np;
proc printto log="&reportlog\&&prog&j...log" new;
run;
%include "&report\&&prog&j...sas";
proc printto;
run;
%end;
***end;
```

CHECK FOR ISSUES IN PROGRAMS AND OUTPUTS

The TFL macro will check for issues in programs and outputs, for example, are programs available, are outputs generated, are there any log issues and are data sets matched. TFL programs, outputs, logs and proc compare reports are located in different folders. The TFL macro will read into SAS the files in different folders and examine them individually. If it is a new or small study you may not need to check if the programs and outputs are created since either they are all new or you can easily identify which programs are missing. However, we found that this process is very useful for an old and large study. With
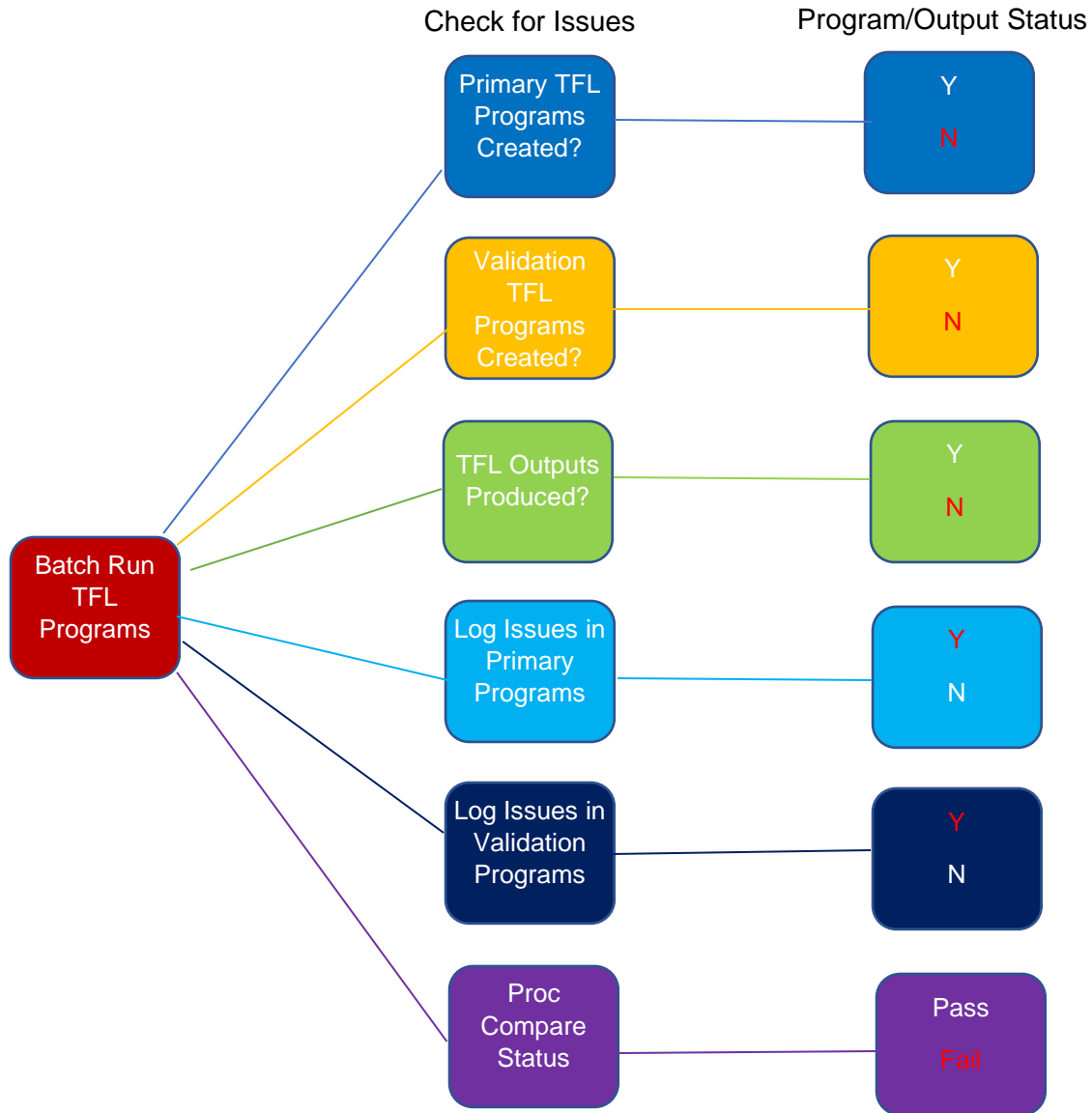
Check for Issues — Program/Output Status

Primary TFL Programs Created? — Y / N

Validation TFL Programs Created? — Y / N

TFL Outputs Produced? — Y / N

Batch Run TFL Programs

Log Issues in Primary Programs — Y / N

Log Issues in Validation Programs — Y / N

Proc Compare Status — Pass / Fail

Figure 1. Contents of the TFL macro

help of this macro you don't need to search a few missing programs from hundreds of TFL programs. It will quickly tell you how many and what new programs are needed to be created for the current delivery. Missing TFL outputs can also be identified using similar programs. Missing TFL outputs can be due to missing programs or programming errors which can be determined by checking if program is missing and there is log issue in program. Log issue checking is kept as simple as possible. The TFL macro will not tell how many log issues and what they are in the program but check if there is any log issue in the program. If yes, we have another log checking macro in the program to find out what and where they are. Log files are created per TFL programs. If one program are involved in generating multiple outputs and it may has log issues related to only one output. The status report will show log issues in all outputs created by the program. Proc compare report is generated per TFL programs as like log files. If one program creates multiple outputs the proc compare report for all outputs will be stored in one file named by program name. The TFL macro will examine the proc compare report by output name and generate status report for each output even they are created by the same program.

GENERATE STATUS REPORT

Although the TFL macro checks all files in program, output, log and proc compare folders only the list of TFLs extracted from TOC will appear in the status report. The issue reports from program, output, log and proc compare will be integrated together by merging with TFL list by either program and output name. Our goal is to make our status report as simple and clear as possible and to allow lead programmer to quickly identify the issues. The status report will appears as 'Y' or 'N', 'Pass' or 'Fail' for each categories. All issues will be marked as 'Y', or 'N' or 'Fail' in color in the status report. N/A means 'Not Applicable'. For example no QC is required for listings as stated in study QC plan, therefore there are no validation log and proc compare report for listings.

TFL Name	Program Name	Primary			Validation		
		TFL Programs Created?	Outputs Produced?	Primary Log Issues?	TFL Programs Created?	QC Status	Validation Log Issues?
FIGURE_01_LAB	f_lab	N	N/A	N/A	N	N/A	N/A
LISTING_01_DISP	l_disp	Y	Y	N	N/A	N/A	N/A
LISTING_05_AEM	l_aem	Y	N	N	N/A	N/A	N/A
TABLE_02_DEMOG	t_demog	Y	Y	Y	Y	Pass	N
TABLE_06_OAE	t_oae	Y	Y	N	Y	Fail	N
TABLE_14_LABS	t_labs	Y	Y	N	Y	Pass	Y

Display 1. The program/Output Status Report

HOW TO USE

The macro is very easy to use. There is no macro variable. You may run it anywhere after study is initiated. It is also easy to copy to a different study if they have the same folder and path naming convention set. If the folder structure and path are different you may just update path inside macro. This macro can be easily updated to use for monitor the status of SDTMs and ADaMs.

CONCLUSIONS

This macro is developed to monitor the program/output status for a large and old study with more than two hundreds of programs and outputs and frequent delivery requests. We find it has been useful for us to manage the resource needs for each delivery by checking the program status with this macro before programming really starts. This macro can also be used to monitor program/output status anytime during the process of programming and to double check the final status before delivery. For a study with more hundreds of TFLs it is tedious and time consuming to check log issues and proc compare status one by one. This macro can result in significant improvement in resource management and program status tracking.

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

Sayeed Nadim
PRA Health Science
721 Arbor Way Suite 100
Blue Bell, PA 19422, USA
nadimsayeed@prahs.com

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