PharmaSUG 2020 - Paper AP-129

Shell Script automation for SDTM/ADaM and TLGs

Pradeep Acharya, Ephicacy Lifescience Analytics Pvt Ltd.;
Aniket Patil, Pfizer Inc.

ABSTRACT

In clinical trials, every study has multiple datasets (SDTM/ADaM) and several number of tables, listings and figures (TLFs). Throughout the study, there are many instances where you are required to re-run the datasets due to new incoming data or updates in key datasets such as subject level dataset (ADSL), which results in refresh of all other datasets and TLFs to maintain timestamp consistency. One of the approaches is to run the datasets and TLFs one after the other consuming a significant amount of time for program execution. Another commonly used method is to manually create an executable batch/script file that includes all the program names. The drawback in these methods is you can inadvertently miss a dataset/TLG program execution due to manual error, and then it becomes cumbersome to go through the logs and outputs to identify which program has been missed. To overcome such situations and reduce repetitive tasks, we have developed an automated program which creates the shell script by reading the program names directly from the designated study folder for all the datasets and/or TLGs, thereby eliminating any chances of missing any program. This program also provides flexibility to decide the order in which your programs will be run as is generally the case with SDTM/ADaM programs. If there is one to one relation between programs and output for TLFs more functionality can be added to cross check the numbers and confirm a smooth run. Having this facility frees up programmer time for more productive work.

INTRODUCTION

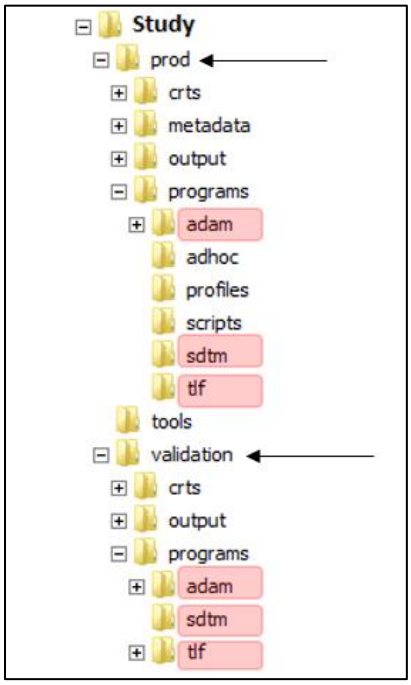
Clinical trials are the studies to evaluate the effectiveness and safety of medications or medical devices by monitoring their effects on large group of people. To assess these objectives, the information is collected in the form of different datasets and the summaries are depicted in the form of reports. The current FDA data standards catalogue specifies the use of CDISC standards for data submission. Every clinical trial has multiple SDTM/ADaM datasets and many numbers of Tables, Listings and Figures (TLFs), which are created through the programming using different software or analysis packages. In this paper, we refer to programs created using SAS® software.

As mentioned earlier, there are many instances where you need to execute the datasets and reports multiple times throughout the study for various reasons like new source data extraction, update in the key datasets, change in derivation rule or addition/deletion of variables etc. Executing these programs one by one or creating an executable file manually, which consists of all the program names, is time consuming and error prone especially where large number of TLFs are involved. In order to simplify this task and minimize human error, we developed a utility macro which creates an executable shell script or batch file consisting of all valid SAS program names in the designated folder of the study. This shell script, a txt file with specific commands, is executed through Unix/Linux. The execution order of the programs within the script can be customized for SDTM and ADaM for any interdependency between datasets. In the following sections we will explain the creation and execution of shell script.

CREATION OF SHELL SCRIPT

Let us consider the study folder structure as depicted in figure 1. The individual SDTM, ADaM and TLF programs reside in their designated folders within the prod and validation sub folders. The utility macro which creates the shell script is placed in each of these folders. The macro identifies the location folder and names the scripts accordingly viz. *shscript_prod_sdtm, shscript_validation_sdtm, shscript_prod_adam, shscript_prod_tlf etc.*

Figure 1.



Now let's look at some important code snippets.

The following snippets of code are part of macro ***%create_shscript (dataord =, dtnames =)***

Snippet 1 code is designed to Identify the location of the folder where the macro is run in order to appropriately name the resulting shell script. It illustrates how the script file name is automatically assigned.

Snippet 1

```
data loc;
  length pgmloc &len.;
  pgmloc = "&_pwd.";
  call symput ("_cwd", pgmloc);

  call symput ("cdir", scan(pgmloc, -3, "/"));
  call symput ("cloc", scan(pgmloc, -1, "/"));

  outname = compress(lowcase("batch_file_&cdir._&cloc."));
  call symput ("outfile", outname);
run;
```

The *&_pwd* macro variable refers to the complete location of the program. This variable is read to identify the current directory of *production* or *validation* in the macro variable *&cdir* and the current *location* of sdtm/adam/tlf is captured in another macro variable *&cloc.* By combining these two variables we can customize the script file name as - shscript_&cdir._&cloc. (*shscript_prod_sdtm.sh, shscript_validation_sdtm.sh, shscript_prod_adam.sh, shscript_prod_tlf.sh etc.)* An extension of *.sh* is added to the files to make them executable in linux/unix.

The code in snippet 2 collects the names of all files (not only *.sas* ext) located in the study folder and creates a sas dataset with one record for each file name. The variable name is *fname.* The intermediate dataset *all* has all the file names (Ex-.sas .log .lst .txt .rtf etc.) from the folder. Next we get rid of all other file formats as we are only interested in keeping sas program names to include in the script.

Snippet 2

```
data all;
  rc=filename("pgms", "&dirr.");
  did=dopen("pgms");
    if did > 0 then do;
       num=dnum(did);
           do i=1 to num;
               fname=dread(did,i);
               output;
            end;
    end;
run;

if index(fname, 'sas') eq 0 then delete;
```

The program folder may have *.sas* files other than the intended dataset/TLF programs we aim to execute (Ex: test.sas, check.sas, autoexec.sas, myprog.sas etc.) The code in snippet 3 keeps only *valid* sas program names. The assumptions are - two letter names for SDTM programs *(dm.sas)*, ADaM programs names starting with *ad (adeff.sas),* TLF programs starting with t_, l_ and f_ respectively *(t_disp.sas, l_ae.sas, f_mfs.sas)*. Correspondingly all the validation program names starting with v_. *(v_dm.sas, v_adeff.sas, v_t_disp.sas).* The only exception is of *checklog.sas* program to check logs of all programs which is set to execute at the end.

Snippet 3

```
fname=upcase(fname);
if substr(compress(fname),3,4) eq ".SAS" then TLF = "SD";
else if substr(compress(fname),1,2) eq "V_" and
substr(compress(fname),5,4) eq ".SAS" then TLF = "VSD";

else if substr(fname,1,2) in ("T_","L_","F_") then TLF = substr(fname,1,1);
else if substr(fname,1,4) in ("V_T_","V_L_","V_F_") then TLF =
compress(upcase("V"||substr(fname, 3, 1)));

else if substr(fname,1,2) eq "AD" then TLF = "D";
else if substr(fname,1,4) eq "V_AD" then TLF = "VD";

else if index(fname, "CHECKLOG") gt 0 then TLF = "C";

if TLF ne " ";
```

The utility macro (%create_shscript) takes 2 parameters as noted above, *dataord*= with values of Yes/No and *dtnames*= with names of datasets in order of execution, which becomes optional when you pass No for *dataord*. In case of TLF, assuming no execution order is needed, when you pass *dataord=No* the macro sorts the program names alphabetically for all the tlfs. when the script file is executed, the programs will be run alphabetically. For SDTM datasets, if you do not need the execution order then the SDTM datasets are sorted alphabetically. In case of ADaM, ADSL is executed prior to all other datasets. Generally, for datasets, the alphabetical order of execution may not be desired, most likely you would need to run at least a couple of datasets in a specific sequence. In such cases, the value of *dataord*= parameter is Yes, *dtnames*= takes the names of the datasets in execution sequence separated by a pipe. Ex: ADSL | ADAE | ADCM | ADMH | ADDTAES etc. or in case of SDTM - DM | EX | AE | SV etc. The program names passed to the parameter are not case or space sensitive, they can be uppercase, lowercase or mixed case. The only requirement is they must be separated by pipe symbol. Additionally, you do not have to enlist all dataset names, just the ones you need a specific execution order for, rest of the dataset program names will be added alphabetically.

ILLUSTRATION OF THE PROGRAM

Figure 2 below considers the production/ADaM folder as an example to illustrate program execution. As you can see, create_shscript.sas upon execution creates shscript_prod_adam.sh

Figure 2.

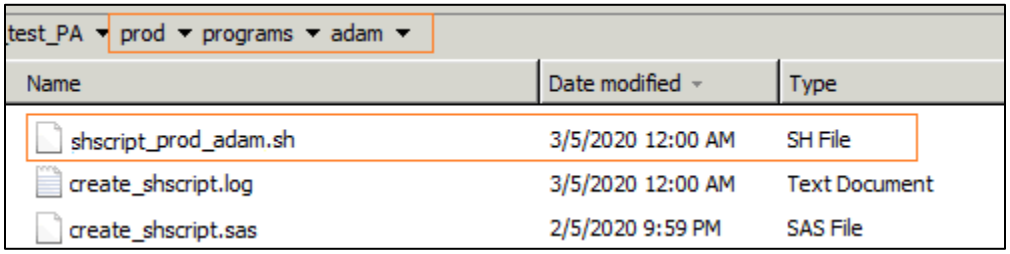


Figure 3 shows the shscript_prod_adam script file. We executed the program with the parameter `dataord=No` ***%create_shscript(dataord=No, dtnames=);*** *adsl.sas* is set to be executed first and then the other dataset programs in alphabetical order with *checklog.sas* at the end of the file. If you notice, we have also included commands to delete the existing *.log* and *.lst* files prior to executing the dataset programs.

Figure 3.

```
#!/bin/bash

#This bash script runs all programs from the current directory
#echo run Production ADAMs programs
rm *.log #
rm *.lst #

gsas adsl.sas #

gsas adae.sas #
gsas adcm.sas #
gsas adlb.sas #
gsas advs.sas #

gsas checklog.sas #
```

When this script file is executed through Unix with the command **sh shscript_prod_adam.sh** all the datasets, *.log* and *.lst* files should have current timestamp as the sample in figure 6 and 7.

Figure 6.

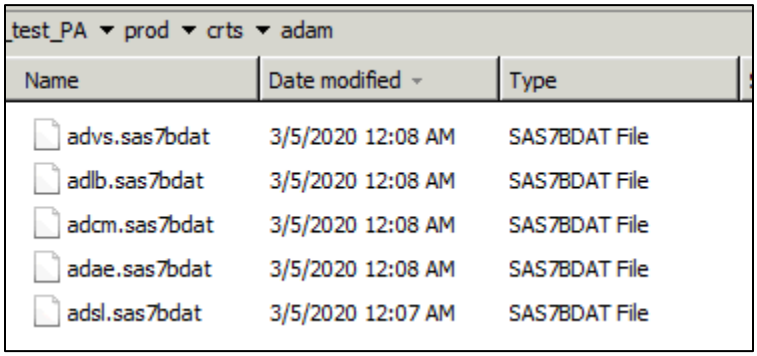
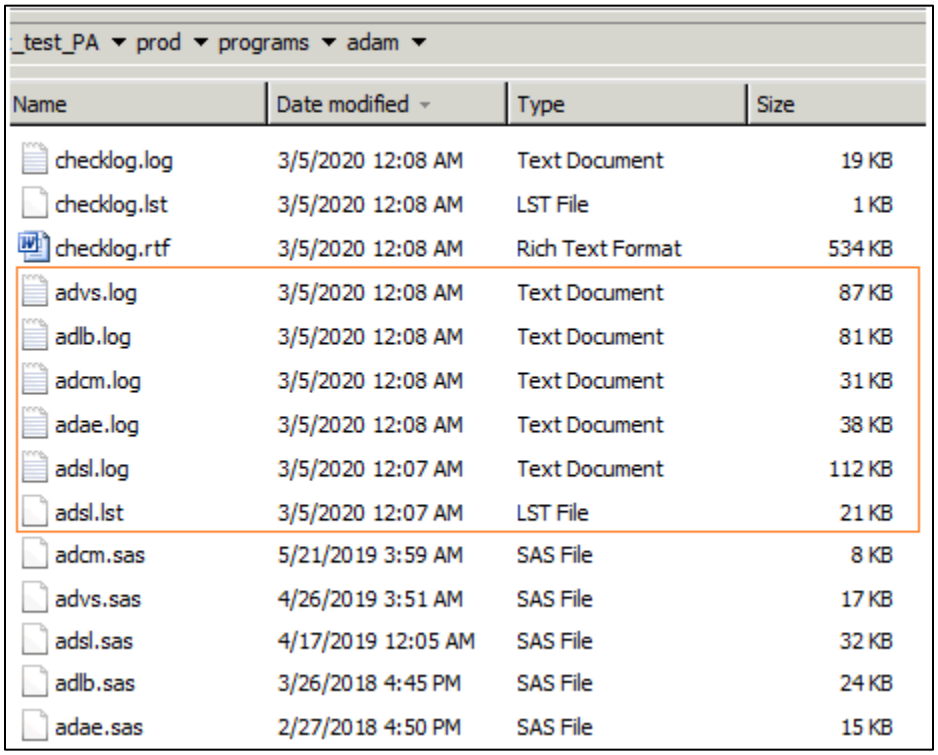


Figure 7.



_test_PA ▾ prod ▾ programs ▾ adam ▾

Name	Date modified	Type	Size
checklog.log	3/5/2020 12:08 AM	Text Document	19 KB
checklog.lst	3/5/2020 12:08 AM	LST File	1 KB
checklog.rtf	3/5/2020 12:08 AM	Rich Text Format	534 KB
advs.log	3/5/2020 12:08 AM	Text Document	87 KB
adlb.log	3/5/2020 12:08 AM	Text Document	81 KB
adcm.log	3/5/2020 12:08 AM	Text Document	31 KB
adae.log	3/5/2020 12:08 AM	Text Document	38 KB
adsl.log	3/5/2020 12:07 AM	Text Document	112 KB
adsl.lst	3/5/2020 12:07 AM	LST File	21 KB
adcm.sas	5/21/2019 3:59 AM	SAS File	8 KB
advs.sas	4/26/2019 3:51 AM	SAS File	17 KB
adsl.sas	4/17/2019 12:05 AM	SAS File	32 KB
adlb.sas	3/26/2018 4:45 PM	SAS File	24 KB
adae.sas	2/27/2018 4:50 PM	SAS File	15 KB

Let's see the case where a specific order of execution is applied.

```
%create_shscript (dataord=Yes, dtnames=adsl |ADcm|adLB| adaE);
```

Figure 8 reflects the script generated, which contains the same execution order for the datasets as the parameter passed. Here, purposely the dataset names were given in different cases (upper case, lowercase, mixedcase, and with/without space), to illustrate that the parameter "*dtnames*" in this program is not case sensitive or space sensitive.

Figure 8.

```
#!/bin/bash

#This bash script runs all programs from the current directory
#echo run Production ADAMs programs
rm *.log #
rm *.lst #

gsas adsl.sas #

gsas adcm.sas #
gsas adlb.sas #
gsas adae.sas #
gsas advs.sas #

gsas checklog.sas #
```

CONCLUTION

Automating a shell script or windows batch file generation saves valuable time for programmers which can be utilized for more productive tasks. The functionality of the utility macro process presented in this paper can be further enhanced to generate a validation report to provide additional details of validation results which can summary of the passed/failed items, specific program names that failed etc. Taking an automated approach to tackle issues related to repeated execution of programs saves a lot of time and effort thereby allowing the teams to focus more energy on delivering quality outputs.

ACKNOWLEDGEMENTS

Authors would like to extend their sincere thanks to Ephicacy and Pfizer for giving them an opportunity to write this paper. Any brand and product names are trademarks of their respective companies.

REFERENCES

Michele. M . Burlew. 2014. *SAS Macro Programming Made Easy*. 3rd ed. SAS Campus Drive, Cary, North Carolina: SAS Institute.

Shell Scripting Tutorial. 2000. “A First Script”, Accessed February 27, 2020. https://www.shellscript.sh/first.html

Bash Reference Manual. 2018. “The GNU Bash Reference Manual” Accessed March 5, 2020. https://tiswww.case.edu/php/chet/bash/bashref.html#Introduction

CONTACT INFORMATION

Pradeep Acharya, Senior Lead Biostatistician, Ephicacy Lifescience Analytics Pvt. Ltd., Bengaluru (email: pradeep.acharya@ephicacy.com)

Aniket Patil, Senior Manager, Pfizer Inc. (email: aniket.patil@pfizer.com)