PharmaSUG 2018 - Paper QT-11

Bring Unix into SAS®

Ying Guo, Ming Yan, Eli Lilly and Company

ABSTRACT

While SAS® is a great tool for clinical programming, it has some limitations. Most programs will work on one single file, and cannot change the folder structure or apply similar changes to a group of files under one directory. Thus, automated programs must be supplemented with manual window actions. With repeated tasks, these manual window actions become time consuming. For example, if you have 50+ programs to move from one location to another, the program path needs to be updated for all programs. Manually updating each file is time consuming. SAS® has the ability to communicate with external software such as Excel, Word and UNIX. UNIX has the power to execute commands to edit folder structure and file content. This paper will describe step-by-step instructions of how to utilize Unix commands within the SAS® environment. Specifically, we will describe the following functionality: (1) Create a new folder or folders; (2) Move/copy/delete a file or files in the target folder; (3) Modify the content of a single file; (4) Modify the same content for all files within one folder simultaneously; (5) Create similar files repeatedly. At the end, we will also list the advantages of using this method vs. the traditional manual method. By using this new method, programmer can significantly increase the efficiency in making programming modifications by reducing the manual windows actions. .

INTRODUCTION

SAS® is a great tool to manipulate the data, conduct statistical analysis, and prepare the fancy outputs. However, there are something that SAS® cannot do alone. At the same time, the operational system have powerful commands, which can achieve numerical goals. Using those commands in SAS® programming can dramatically increase SAS® ability, and maximize automation. Luckily, SAS® has one macro statement, %sysexec, which allows users to go into operating environment mode under most operating environments, where you can issue operating environment commands and return to your SAS® session. This paper will provide examples of how to use UNIX command in SAS®, and give step-by-step instructions on how to achieve this under SAS® environment. UNIX platform is required to use the following methods.

COMMUNICATED WITH UNIX IN SAS®

%SYSEXEC is a macro statement, which puts you into operating environment mode under most operating environments, where you can issue operating environment commands and return to your SAS® session. This statement will allow you to use UNIX command in SAS®. If commands contain a semicolon, use a macro quoting function. The following are examples of how to achieve certain command in SAS®.

CREATE FOLDER IN SAS®

During SAS® programming, sometimes you'll feel that you would like the output go to a new folder, instead of bundling with other outputs. Sometimes, you'd like to have a consistent folder across the entire therapeutic area, and would like to create the same folder structure for all different studies. Manually created each folder will be time consuming. Using %sysexec combine with mkdir UNIX command can achieve this goal.

1. Create a new folder: If you already have a path as /lillyce/qa/<compound>/<study>/output/, and would like to create a new folder under output, you can use the following command. "mkdir" is the command to instruct UNIX create new folder. "-p" is the option that UNIX will create all folders on the path. For example, if your folder structure has only /lillyce/qa/<compound>/, the following command will create all <study>, output, and safety folders.

```
%sysexec mkdir -p /lillyce/qa/<compound>/<study>/output/safety/;
```

2. You can also write a macro, and create folders for different compound or studies, to ensure the consistence of the folder structure. Here is an example.

   ```
   %macro setfolder(compound=, study=);
   %sysexec mkdir -p
   /lillyce/&area./&compound./&study/programs/analysis/primary/;
    %sysexec mkdir -p
   /lillyce/&area./&compound./&study/programs/analysis/validation/;
    %sysexec mkdir -p
   /lillyce/&area./&compound./&study/programs/observed/primary/;
    %sysexec mkdir -p
   /lillyce/&area./&compound./&study/programs/observed/validation/;
    %sysexec mkdir -p
   /lillyce/&area./&compound./&study/programs/tfl/primary/;
    %sysexec mkdir -p
   /lillyce/&area./&compound./&study/programs/tfl/validation/;
   %mend;
   ```

COPY OR MOVE A FILE

During SAS® programming, sometimes you'll feel that you would like save the original files before you start to modify it. Using %sysexec combined with cp UNIX command can achieve this goal. IN following SAS® code, "cp" is the command to instruct UNIX copy a file. "-p" means to copy the property of the file. Using –p will retain the original file property, such as file creation time etc. Without this option, all copied files will have the date and time when the file is copied. Path1 is where the document are copied from, and path2 is where the document will be copied to. If the <new name> is missing, then the file will be copied to the new location with the same name. If you would like to move a file, then change "cp" command to "mv". If you would like to delete a file, then use "rm" command.

```
%sysexec cp –p <path1>.ae.sas <path2>.<new name>;

%sysexec mv <path1>.ae.sas <path2>;

%sysexec rm <path1>.ae.sas;
```

Here is an actual example of saving the original copy with the date assigned.

```
%let dummy=/lillyce/qa/xxx-xx-xxxx/prelock/data/raw/shared;
%let ver=_02122018;

/*create new folder and copy the original data*/;
%sysexec mkdir &dummy./original;
%sysexec cp -p &dummy./clrm_lab.sas7bdat
&dummy./original/clrm_lab&ver..sas7bdat;
```

MODIFY CONTENT OF A FILE/FILES

Sometimes, you'd like to have file content modified for other use. For example, you may want to have a file moved from one study to another study, and would like to change the header of the file to reflect the new location. It will be easier to open the file and modify it, if that needs to be done only once. However, it will be really convenience to create a macro, if that needs to be done multiple times. Following command "sed" will help to modify the content. "-i" means after modify the content (don't understand); the file will be saved with new change. "-e" enables multiple changes in one command. In the following example, the command change the file ae.sas. Use ae1 replace ae in the entire document. Since it is case sensitive, two modification commands are included for both lower and upper case.

```
/*modify the file*/;
%sysexec sed -i -e 's/ae/ae1/' -e 's/AE/AE1/' &path.ae.sas;
```

Sometimes, you'd like to change all files in one folder for the same modification. It will be really time consuming if you have to modify it one by one individually. The following is an example of code to modify all SAS® program in one folder. "sed" command can apply the same modification to all files in one folder. "cd" command will move the operation system to designed folder; "find" indicates that only .sas file will be modified; "s/amaf/amag/" indicates that all "amaf" will be replaced by "amag"; "/%include/" means only the line has "%include" will be modified.

```
%sysexec %str(cd //lillyce/qa/programs/observed/validation/; for file in
$(find . -name "*.sas");
do sed -i -e 's/amaf/amag/'
          -e 's/sdtm_spec1.xlsx/sdtm_spec2.xlsx/'
          -e '/%include/ s/zmsetup_pre/zmsetup_prelock/'
      $file;
done;);
```

CREATE REPEATED FILES

Sometimes, you'd like to create similar files. Using "cp" and "sed" can create a macro and create repeated file quickly. The following is an example to create the SDTM calling programs. The only difference among all files are the domain name. Buy copy each file (?), and open in SAS® to modify the domain name will be time consuming. Following steps will create repeated file in seconds.

```
/****create calling programs*/;
%let path=//lillyce/qa/intrm1/programs/observed/validation/;
%let domain=vs;
/*copy the file*/;
%sysexec cp &path.ir_ae.sas &path.ir_&domain..sas
/*modify the file*/;
%sysexec sed -i -e 's/ae/vs/' -e 's/AE/VS/' &path.ir_&domain..sas;
```

USAGE EXAMPLE

%sysexec is not used independently. It can be used within SAS® code. That will help to reduce a lot of manual work on daily basis.

1. Archive: When raw data are updated, you would want to rerun all your programs to refresh the SDTM, ADaMs and TFLs. Sometimes, you'd like to compare the updated TFLs with the old version. By creating archive procedure using "mkdir", "cp" command, you can save the log and SDTM, ADaMs or TFLs before you update them. This will apply this archive procedure (?), without manually coping files for each iteration.

2. Change header/path when moving programs from one compound/study to another. Using "cp" and "sed" you can create a macro to move files and modify the files. In each study, there are usually several interims involved. All interims will use the programs to produce SDTMs, ADaMs and TFLs. Sometimes, not all programs are located in one place. This requires repeated manual work for each interim analysis. A macro combining all steps together will dramatically reduce manual work. This will make it more efficient.

3. When moving or copying a file with big size, such as lab data, copying in window will take a long time. Use 'cp' in unix, will only take one second. This will save a lot of time for big datasets.

CONCLUSION

In summary, UNIX commands have a lot of features which benefit our daily work. Using it in SAS® environment can make SAS® more efficient. This will reduce huge amount of manual work, and improve our SDTM, ADaM and TFL creation process. This paper introduces a concept on how to use platform commands in SAS®. There are more commands available in UNIX, and other platform. User can explore more in this area, and improve their process.

CONTACT INFORMATION

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

Ying Guo
Eli Lilly and Company
317-6559428
Guo_ying@lilly.com