SESUG Paper 101-2018

What's the Difference? - Comparing SAS® Artifacts After a Migration or Upgrade

Brian Varney, Experis Business Analytics Practice

ABSTRACT

There are times when it is convenient to be able to compare two file system areas that are supposed to be equivalent. This could be during a migration activity involving changing servers and/or SAS® versions. This paper intends to cover techniques that can help facilitate the comparison of programs, logs, excel files, and SAS data sets as well as techniques for digesting the results.

INTRODUCTION

This paper intends to address topics regarding the development of SAS code to attempt to automate verification of SAS artifacts during a migration and/or upgrade. We will first frame the problem this paper intends to address, then we will move on to the planning and preparation, and finally we will discuss possible approaches to solve the problem. The objective of this paper is to introduce data driven techniques resulting in more efficient solutions and less tedious manual effort. Therefore, this paper will include retrieving information from file systems and the SAS Macro Language.

FRAMING THE PROBLEM

As stated in the introduction, this paper intends to address topics regarding the development of SAS code to attempt to automate verification of SAS artifacts during a migration and/or upgrade. A migration or upgrade could include a change in hardware and/or software related to the SAS computing environment.

The industry involved will probably drive how formal and stringent the verification and validation of the results before and after a migration or upgrade. SAS environments that are used for regulated industries such as pharmaceutical and financial often have extensive requirements on how the verification and validation is performed and documented.

There are many possibilities as to why you may need to perform migration and/or upgrade of your SAS environment. A few possible scenarios may include but are not limited to:

- Moving to a different physical or virtual server
- Moving to a different operating system or operating system version
- Moving to a newer SAS version or service pack

Regardless of the scenario, it is very likely that extensive verification is required to ensure the target(new) environment is functioning and producing results equivalent to the source(old) system.

If you are in a position of needing to perform extensive verification after a change in environment, the first thing you will want to do is define the scope of what you need to compare. Below is a list of typical SAS artifacts that will need to be compared.

- SAS Programs
- SAS Logs
- SAS Data Sets
- SAS Listing Files
- Miscellaneous ASCII Files

- Microsoft Excel Spreadsheets
- Microsoft Word Documents (tables, listings, figures)

At this point, it may be prudent to review the suggested steps for preparing, copying, and comparing the relevant artifacts. This will transition us to the strategy section.

PLANNING AND PREPARATION

Planning and preparation are key during a migration and/or upgrade. If the right sequence of steps is taken, it can make it easier to implement data driven techniques for the verification process. We will now outline a suggested sequence of steps for a migration and/or upgrade.

ON THE SOURCE SYSTEM (PRIOR TO MOVING/COPYING PROJECT FOLDERS TO THE TARGET ENVIRONMENT)

1. Schedule a project "suspension period" during which users stop working with the project files and folders.
    a. It is beneficial to have upper level management support on this.
2. Determine the "completeness" of verification/validation. There should be agreement on how extensively the results are verified. And the % of the results that are verified. i.e. 100% of SAS Data Sets, 100% of summary tables, 25% of detailed listings, and 75% of figures/graphics.
    a. This needs to be documented ahead of time and should have sign off by upper management.
3. Clean up unneeded project artifacts, folders, etc.
4. Identify folders that can be ignored after being copied to the target system.
5. Re-run all programs that will need to have the results verified/validated.
6. Copy the entire project folder into a backup folder.
7. Users need to leave the project files and folders alone until after the verification is finished.

COPY THE ENTIRE PROJECT FOLDER AND BACKUP FOLDER TO THE TARGET ENVIRONMENT

1. These copies are often time consuming and are scheduled in the off hours.
2. All file and folder date time stamps should be kept intact.
3. Users need to leave the project files and folders alone until after the verification is finished.

EXECUTION

This is where the target environment is proven to produce equivalent results to the source environment. It is vital that the users do not try to work in the target environment until the verification/validation activities are complete.

BEGIN VERIFICATION/VALIDATION ACTIVITIES ON THE TARGET ENVIRONMENT

1. Make any modifications necessary and run the relevant SAS programs in the target environment.
    a. Configuration file changes
    b. Autoexec file changes
    c. Driver program changes
2. Compare raw input data sets, Microsoft Excel Spreadsheets, etc.
    a. Document any differences and potential impacts downstream
    b. Find a solution to make them match
    c. Document activities and results
3. Run subsequent builds in the order that they run and compare the data sets
    a. Document any differences and potential impacts downstream
    b. Find a solution to make them match
    c. Document activities and results
4. Repeat until all data set build phases are complete and data sets match
    a. SDTM
    b. ADaM
    c. Any other SAS ETL processes that provide data to output such as tables, listings, and figures
    d. Document activities and results
5. Begin running tables, listings, and graphs and compare output.
    a. A change in SAS version will likely result in formatting differences
    b. If timelines are tight, you can run these prior to running data set builds and focus on changes that need to be made to SAS ODS code. Then re-run again for final verification after the data sets have been rebuilt on the target system. This will help get the formatting issues taken care of in parallel to the data set verification.
    c. Document all activities and results

APPROACHES TO THE PROBLEM

There are a wide range of approaches to the problem of comparing the data, output, and results after a migration/upgrade. Building in automation can make the verification/validation activities less tedious, minimize human error, and ease the documentation process.

MINIMIZING THE MIGRATION/UPGRADE IMPACT

In planning ahead for the migration/upgrade verification/validation activities, there are certain decisions that can be made to minimize the migration/upgrade impact.

- Minimize the number of changes during the migration/upgrade.
    - Keep data set, table, listing, and graph program changes to only what is required to get the program to run in the target environment.
    - If possible, keep the directory structure of the source and target environments the same (or as similar as possible).

- Do not update the raw data between the time the backup is taken and the completion of the verification/validation.

- Cleaning up unnecessary files
  - It seems like nobody ever has time for this. As we all know, project directories often have many unnecessary copies, backups, dead code, etc. Getting rid of all of this prior to the migration/upgrade helps immensely.

METHODS FOR AUTOMATING PIECES OF THE VERIFICATION/VALIDATION ACTIVITIES

It would be nice to push a button and make the verification/validation process happen. Although it may be possible, it would most likely require more effort than makes sense. However, the SAS programming language has the features and functionality to help automate some of the verification/validation activities. We will discuss some methods that take advantage of some of these features and functionality next.

The Backup

As stated earlier, taking a backup of the project directory just after re-running the programs in the source environment in the proper order is crucial. Of course, we need to compare the before and after to effectively verify/validate the target environment results. It is best if the backup is complete and keeps the same directory structure. For example, if the project data, artifacts are stored in …

/projectdata/phase1/study_8675309/ just copy everything to
/projectdata/phase1/study_8675309/BACKUP

The Inventory

The inventory is key to automation. The inventory of the file system and saving it as a SAS data set is an activity that much of this paper depends on. This can be accomplished regardless if your environment is Windows or UNIX/Linux based. If your SAS programming environment does not allow you to submit "X commands" in your program, you may have to generate the inventory file from the command line of the server. Depending on the exact version and flavor of your operating system, the following commands may differ slightly.

UNIX/Linux Command Line: ls -laR "*projectdirectory/*" > "inventoryoutputlocation.*txt*"

Sample program to read in command results:

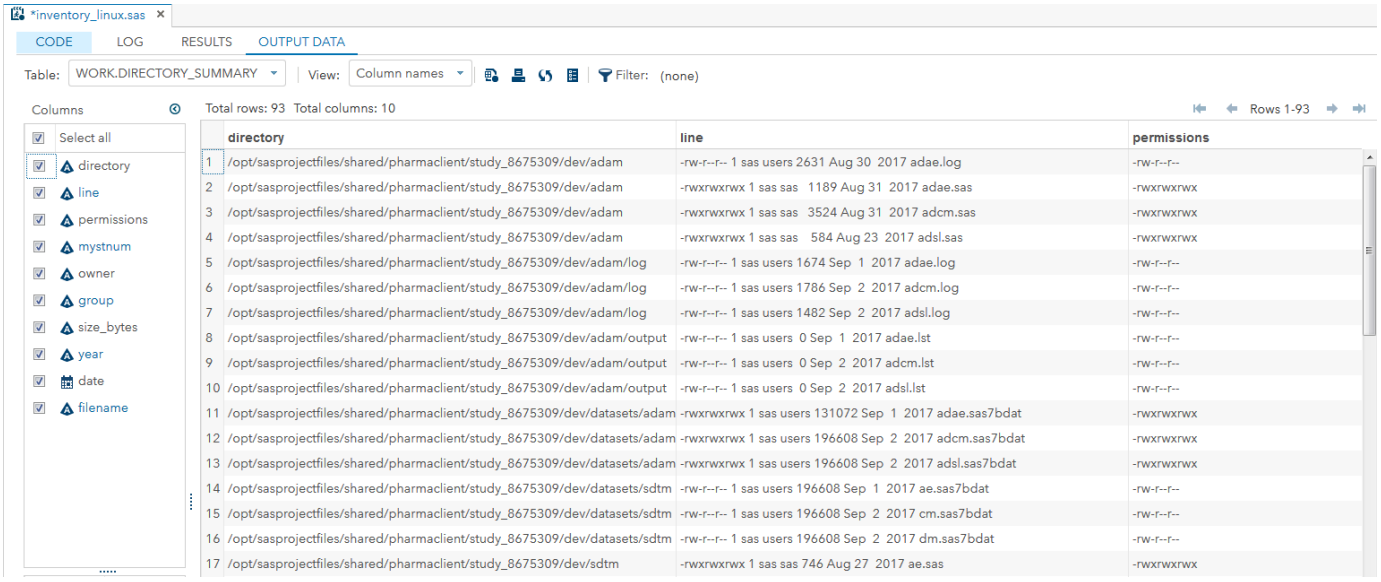
```
** keep data set smaller on disk since it has lots of character variables **;
options compress=yes;


** if X commands are allowed in programs, use the pipe technique to read the
command
** results into a data set. Otherwise, read in the ASCII file from the
command line results.
**;
```

```
filename pipedir pipe ' ls -laR /opt/sasprojectfiles/shared/pharmaclient'
lrecl=5000;


data directory_summary;
   infile pipedir truncover;
   retain directory;
   input line $char5000.;
   if line NE ' ';
   if substr(lowcase(line),1,5) ne 'total';
   if line=:'/' and substr(line,length(line),1)=':' then
   do;
      directory=substr(line,1,length(line)-1);
   end;
   else if substr(line,1,1)='d' then
   do;
   end;
   else do;
      permissions=scan(line,1,' ');
      mystnum=scan(line,2,' ');
      owner=scan(line,3,' ');
      group=scan(line,4,' ');
      size_bytes=scan(line,5,' ');
      year=scan(line,8,' ');
      if index(year,':') then year=put(year("&sysdate9."d),best12.);
      date=input(trim(left(scan(line,7,' ')))||trim(left(scan(line,6,' ')))||
          trim(left(year)),date9.);
      filename=scan(line,-1,' ');
      output;
   end;
   format date mmddyy10.;
run;
```

Display 1 shows an excerpt from output data set from the data step above.



*inventory_linux.sas

CODE | LOG | RESULTS | OUTPUT DATA

Table: WORK.DIRECTORY_SUMMARY | View: Column names | Filter: (none)

Columns: Select all, directory, line, permissions, mystnum, owner, group, size_bytes, year, date, filename

Total rows: 93 Total columns: 10 | Rows 1-93

	directory	line	permissions
1	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam	-rw-r--r-- 1 sas users 2631 Aug 30 2017 adae.log	-rw-r--r--
2	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam	-rwxrwxrwx 1 sas sas 1189 Aug 31 2017 adae.sas	-rwxrwxrwx
3	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam	-rwxrwxrwx 1 sas sas 3524 Aug 31 2017 adcm.sas	-rwxrwxrwx
4	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam	-rwxrwxrwx 1 sas sas 584 Aug 23 2017 adsl.sas	-rwxrwxrwx
5	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam/log	-rw-r--r-- 1 sas users 1674 Sep 1 2017 adae.log	-rw-r--r--
6	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam/log	-rw-r--r-- 1 sas users 1786 Sep 2 2017 adcm.log	-rw-r--r--
7	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam/log	-rw-r--r-- 1 sas users 1482 Sep 2 2017 adsl.log	-rw-r--r--
8	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam/output	-rw-r--r-- 1 sas users 0 Sep 1 2017 adae.lst	-rw-r--r--
9	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam/output	-rw-r--r-- 1 sas users 0 Sep 2 2017 adcm.lst	-rw-r--r--
10	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/adam/output	-rw-r--r-- 1 sas users 0 Sep 2 2017 adsl.lst	-rw-r--r--
11	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/datasets/adam	-rwxrwxrwx 1 sas users 131072 Sep 1 2017 adae.sas7bdat	-rwxrwxrwx
12	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/datasets/adam	-rwxrwxrwx 1 sas users 196608 Sep 2 2017 adcm.sas7bdat	-rwxrwxrwx
13	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/datasets/adam	-rwxrwxrwx 1 sas users 196608 Sep 2 2017 adsl.sas7bdat	-rwxrwxrwx
14	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/datasets/sdtm	-rw-r--r-- 1 sas users 196608 Sep 1 2017 ae.sas7bdat	-rw-r--r--
15	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/datasets/sdtm	-rw-r--r-- 1 sas users 196608 Sep 2 2017 cm.sas7bdat	-rw-r--r--
16	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/datasets/sdtm	-rw-r--r-- 1 sas users 196608 Sep 2 2017 dm.sas7bdat	-rw-r--r--
17	/opt/sasprojectfiles/shared/pharmaclient/study_8675309/dev/sdtm	-rwxrwxrwx 1 sas sas 746 Aug 27 2017 ae.sas	-rwxrwxrwx

Display 1. Excerpt of Output Data Set from Inventory of Linux Study Directory

Windows Command Line: dir "*projectdirectory/*" /S /Q > "inventoryoutputlocation.*txt*"

Sample program to read in command results:

```
** keep data set smaller on disk since it has lots of character variables **;
options compress=yes;

** if X commands are allowed in programs, use the pipe technique to read the
command
** results into a data set. Otherwise, read in the ASCII file from the
command line results.
**;
filename pipedsq pipe ' dir "C:\_mystuff\sas_stuff\sesug_2018" /S /Q'
lrecl=5000;

data pipedirin;
   infile pipedsq truncover;
   input line $char1000.;
run;

data pipedir;
   length directory $1000;
   retain directory;
```

```
   set pipedirin;
   if line =' ' or
      index(upcase(line),'<DIR>')   or
      left(upcase(line))=:'VOLUME'  then
     delete;
   if left(upcase(line))=:'DIRECTORY OF' then
   directory=left(substr(line,index(upcase(line),'DIRECTORY OF')+12));
   if left(upcase(line))=:'DIRECTORY OF' then
      delete;
   if input(substr(line,1,10),?? mmddyy10.) = . then
      substr(line,1,10)='12/31/2999';
   date=input(substr(line,1,10),?? mmddyy10.);
   format date mmddyy10.;
run;

proc sort data=pipedir;
   by directory descending date;
run;

data Directory_Summary(drop=i line);
   set pipedir;
   by directory;
   length filename $200;
   retain directory_size;
   if first.directory then
   do;
     directory_size=input(scan(line,4,' '),comma32.)/1000000;
   end;
   file_size=input(scan(line,4,' '),comma32.)/1000000;
   file_owner=scan(line,5,' ');
   filename=' ';
   do i=6 to 100;
      filename=trim(left(filename))||' '||scan(line,i,' ');
     if scan(line,i,' ')=' ' then
        leave;
   end;
   if index(filename,'.') then
```

```
      extension=scan(lowcase(filename),-1,'.');
   else
      extension=' ';
   if index(upcase(line),'FILE(S)') then
      delete;
   if date ge '30DEC2999'd then
      delete;
run;
```

Uses for the Inventory Data Set

Once your file system data regarding the project data is saved into a SAS Data Set, there are many uses related to activities and processes surrounding the verification/validation process.

1. Searching for locations in code that require modification for the target environment. You can search and replace but do so at your own risk. A few examples follow:
    a. options
    b. library references
    c. filename references
    d. programming elements that need to be changed because of the newer SAS version.
2. Comparing errors and warnings between the active and backup logs
3. Comparing SAS Data Sets between active and backup directories (PROC COMPARE)
4. Comparing ASCII files between active and backup directories
5. Comparing MS Excel files between active and backup directories

CONCLUSION

Using the techniques and processes in this paper can drastically reduce the manual effort and highlight differences between source and target environments quickly. All environments vary but some automation is typically valuable and available.

ACKNOWLEDGMENTS

I would like to acknowledge David Bosak and Srinivas Dhara for their contributions during past projects involving migrations and upgrades.

CONTACT INFORMATION

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

Brian Varney
Experis Business Analytics
269-553-5185
brian.varney@experis.com
www.experis.us/analytics
www.experis.us/clinical