PharmaSUG 2015 - Paper QT41

Automated Checking Of Multiple Files

Kathyayini Tappeta, Percept Pharma Services, Bridgewater, NJ

ABSTRACT

Most often clinical trial data analysis has tight deadlines with very close data transfers. When last minute modifications are done by different team members, an automated check of the required content in several files ensures increased efficiency under tight timelines. This obviates the need for each file to be opened and checked. The basic concept for such checks involves opening multiple files at a given location, extracting information from these to a single file and performing required checks so that each individual file need not be opened. It can have several applications including checking multiple log files for errors/ warnings, checking multiple proc compare files for notes and in case of macros written for tables, listings and graphs the parameter values can be checked to see if entered correctly. This paper provides code for automating the checking of files with example applications.

INTRODUCTION

Multiple file checking is a common task in clinical trial data analysis. Most companies have dataset and TLF output (log, proc compare, parameter entry) files stored in a folder. A general task done with data transfers is to re-run the programs with the new data. Sometimes there may be a change that has to be implemented across outputs. In either case once the final run is done, a programmer may want to do a high level check to see if the correct changes were implemented or look for specific information across files. Some such checks of interest would be:

- macro parameter entry files: Check if a correct parameter value has been entered across the outputs or if the correct change has been made across a similar set of outputs
- proc compare files: Check for mismatches and other comparison information
- log files: Check the file for errors/warning/specific notes

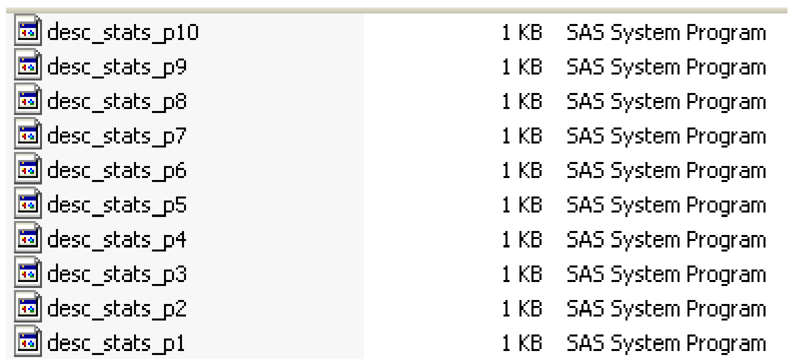
In such situations, opening each file and checking is time consuming and prone to errors. A more efficient way would be to bring in the required information to one file and then check this file. This will decrease the rate of error, take lesser time, and can act as an added layer of check.

CHECKING MACRO PARAMETER ENTRY FILES

A macro will be presented below to automate the process of extracting information from a set of files. The below macro is an example to check values entered for parameters across a set of outputs. This could be used as a screening quality control check or anywhere along the process when you need to quick glance to double check.

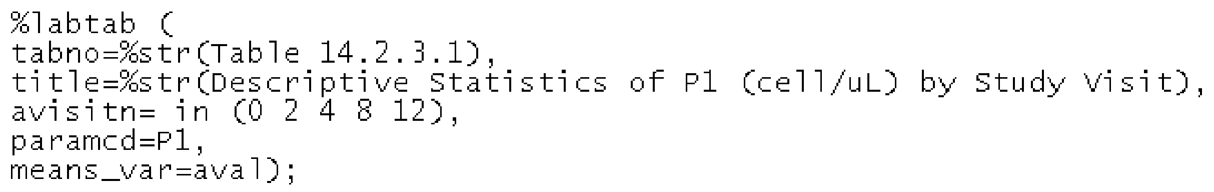
Consider a macro written for 10 descriptive statistics tables. The macro parameter value changed for each case are, titles, footnotes, laboratory parameter value and the analysis result used to calculate the mean. The programmer first creates copies of the macro parameter entry file and then changes the value for a these parameters. You want to ensure that with the duplication of the macro parameter entry file, the parameter has been changed correctly. You want to check that the file name, title number, title, lab parameter value are all consistent.

Consider the below as the set of files (Display 1) that you want to extract information from and the contents of one of the file (Display 2).



desc_stats_p10	1 KB	SAS System Program
desc_stats_p9	1 KB	SAS System Program
desc_stats_p8	1 KB	SAS System Program
desc_stats_p7	1 KB	SAS System Program
desc_stats_p6	1 KB	SAS System Program
desc_stats_p5	1 KB	SAS System Program
desc_stats_p4	1 KB	SAS System Program
desc_stats_p3	1 KB	SAS System Program
desc_stats_p2	1 KB	SAS System Program
desc_stats_p1	1 KB	SAS System Program

Display 1. Macro parameter entry files in the folder



```
%labtab (
tabno=%str(Table 14.2.3.1),
title=%str(Descriptive Statistics of P1 (cell/uL) by Study Visit),
avisitn= in (0 2 4 8 12),
paramcd=P1,
means_var=aval);
```

Display 2. Contents of macro parameter entry file desc_stats_p1.sas

```
%macro expr (path=, fname=, pname=);
```

Macro parameters required to be entered for this task would be:

&path. → the folder location

&fname. → the files that you want to extract information from

&pname. → the parameters whose values you want to extract (&pname.).

```
count=put(countw("&fname.", " "), best.);      ...... (1)
call sympurt ("cnt", count);

%do i=1 %to &cnt;  ...... (2)
data param&i.;
 length line $200.;
 infile "&path.%sysfunc(scan(&fname., &i, " "))" length=len;  ...... (3)
 input @01 line $varying200. len; ...... (4)
 if _n_>=2;
 filename=scan("&fname", &i, " ");
 var_name=scan(line, 1, "=");                            ...... (5)
 txt=substr(line, index(line,"=")+1);
 if var_name in (&pname);
run;

proc append base=param data=param&i; ...... (6)
run;
%end;

proc transpose data=param out=paramt (drop=_NAME_);  ...... (7)
  by filename;
  var txt;
  id var_name;
run;
```

(1) Counting number of files entered by user

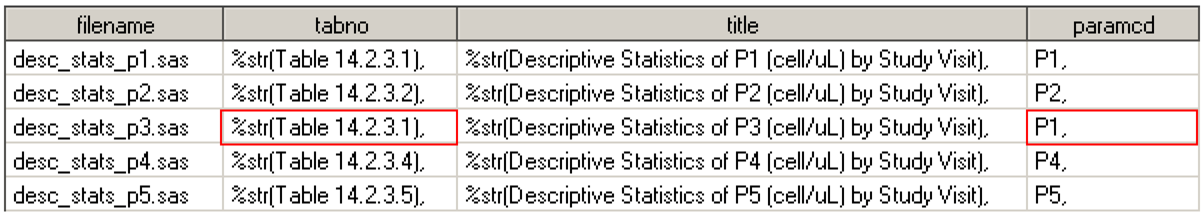
(2) Running a do loop to read in files one at a time

length= option in infile statement to determine length of each record

(3) $varying200. Informat to read the values records with varying length
(4) Extract the filename and values assigned to parameters specified by the user
(5) Proc append to add data from all files into 1 dataset
(6) Transposing to have once record for each file with the macro parameters as variables. Output dataset from the macro is shown below

Submitting the below statements creates the output shown in Display 3. By examining the output and comparing it against the List of Tables, you can see that the table number and parameter name for desc_stats_p3.sas is wrong. This is easier to catch having all the information in one place.

```
%expr (path=%str(C:\Documents and Settings\KTappeta\Desktop\QC\),
       fname=%str(desc_stats_p1.sas desc_stats_p2.sas desc_stats_p3.sas
                  desc_stats_p4.sas desc_stats_p5.sas),
       pname=%str("tabno" "title" "paramcd"));
```

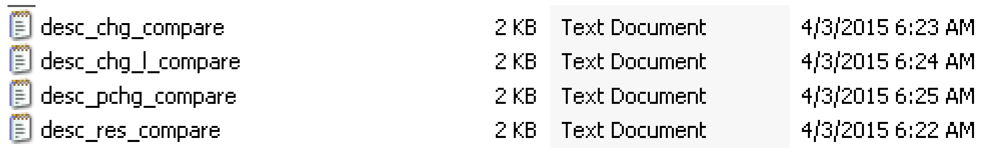


filename	tabno	title	paramcd
desc_stats_p1.sas	%str(Table 14.2.3.1),	%str(Descriptive Statistics of P1 (cell/uL) by Study Visit),	P1,
desc_stats_p2.sas	%str(Table 14.2.3.2),	%str(Descriptive Statistics of P2 (cell/uL) by Study Visit),	P2,
desc_stats_p3.sas	%str(Table 14.2.3.1),	%str(Descriptive Statistics of P3 (cell/uL) by Study Visit),	P1,
desc_stats_p4.sas	%str(Table 14.2.3.4),	%str(Descriptive Statistics of P4 (cell/uL) by Study Visit),	P4,
desc_stats_p5.sas	%str(Table 14.2.3.5),	%str(Descriptive Statistics of P5 (cell/uL) by Study Visit),	P5,

Display 3. Output dataset after running the macro to check macro parameter entry files

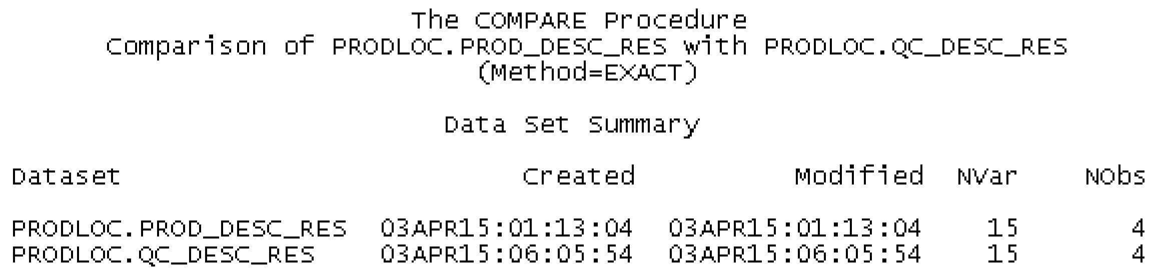
CHECKING PROC COMPARE FILES

Suppose you want to check a few proc compare files to ensure QC was run after the final production run. Also you would like to check that basic dataset summary matches for both production and QC datasets. Consider the below proc compare files in your folder (Display 4) with contents shown in Display 5.



desc_chg_compare	2 KB	Text Document	4/3/2015 6:23 AM
desc_chg_l_compare	2 KB	Text Document	4/3/2015 6:24 AM
desc_pchg_compare	2 KB	Text Document	4/3/2015 6:25 AM
desc_res_compare	2 KB	Text Document	4/3/2015 6:22 AM

Display 4. Proc compare files in the folder



```
                        The COMPARE Procedure
     Comparison of PRODLOC.PROD_DESC_RES with PRODLOC.QC_DESC_RES
                           (Method=EXACT)

                          Data Set Summary

 Dataset                       Created           Modified  NVar    NObs

 PRODLOC.PROD_DESC_RES  03APR15:01:13:04  03APR15:01:13:04    15       4
 PRODLOC.QC_DESC_RES    03APR15:06:05:54  03APR15:06:05:54    15       4
```

Display 5. Dataset summary part of the proc compare output (desc_res_compare.txt)

```
%macro expr (path=, fname=);
```

Macro parameters required to be entered for this task would be:

&path. → the folder location

&fname. → the proc compare files that you want to extract information from (should be .txt files)

```
data chk;
  count=put(countw("&fname.", " "), best.);
  call symput ("cnt", count);
run;

%put &cnt;

%do i=1 %to &cnt;
data param&i. (keep=filename);
 length line $200.;
 retain n;
 infile "&path.%sysfunc(scan(&fname., &i, " "))" length=len;
 input @01 line $varying200. len;
 filename=scan("&fname", &i, " ");
 if line ne "";
 if strip(line)="Data Set Summary" then n=1;
 else n=n+1;
 if n in (3 4);                                                          …… (1)
 if index(line, "QC")>0 then call symput("varqc", line);
 else if index(line, "PROD")>0 then call symput("varp", line);
 if n=4;
run;

%put &varp.;
%put &varqc.;

data param&i._;      …… (2)
  length locationp locationqc $10.;
  set param&i.;
  locationp="%scan(&varp., 1)";
  locationqc="%scan(&varqc., 1)";
  createdp=input("%scan(&varp., 2, " ")", datetime.);
  createdqc=input("%scan(&varqc., 2, " ")", datetime.);
  modifiedp=input("%scan(&varp., 3, " ")", datetime.);
  modifiedqc=input("%scan(&varqc., 3, " ")", datetime.);
  nvarp=input("%scan(&varp., 4, " ")", best.);
  nvarqc=input("%scan(&varqc., 4, " ")", best.);
  nobsp=input("%scan(&varp., 5, " ")", best.);
  nobsqc=input("%scan(&varqc., 5, " ")", best.);
  format createdp createdqc modifiedp modifiedqc datetime.;

run;

proc append base=param data=param&i._;
run;
%end;

data paramt;
  set param;
  length comment $20;
  if locationp ne locationqc or
     modifiedp>modifiedqc or
     nvarp ne nvarqc or
     nobsp ne nobsqc then comment="Check Files";   …… (3)
  else comment="OK";
run;

%mend expr;
```

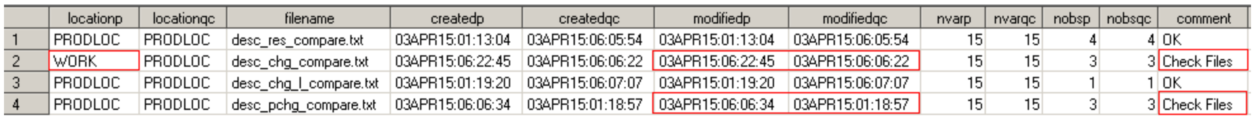
(1) Keep dataset summary information alone from the proc compare output and store production and QC information in two macro variables

(2) Use the production and QC information macro variable (&varp., &varqc.) to create a dataset containing values for each attribute of the dataset (Dataset location, Creation date, Modification date, Number of variables, Number of observations)

(3) Check QC information such as Production dataset was used from original location and no modification was done, QC was run after final production run, number of observations and variables are same in both QC and production datasets

Submitting the below statements creates the output shown in Display 6. By examining the output you can see that in Row 2, production file was brought into the work folder and then compared. This should not be done and the reason must be investigated. In Row 4 you can see that production dataset was modified after the last QC and hence the proc compare output is no longer valid. QC must be done again for this output. Such details are often missed and having the information in one place and automating checking is an efficient alternative.

```
%expr (path=%str(C:\Documents and Settings\KTappeta\Desktop\QC\),
       fname=%str(desc_res_compare.txt desc_chg_compare.txt  desc_chg_l_compare.txt
desc_pchg_compare.txt));
```



	locationp	locationqc	filename	createdp	createdqc	modifiedp	modifiedqc	nvarp	nvarqc	nobsp	nobsqc	comment
1	PRODLOC	PRODLOC	desc_res_compare.txt	03APR15:01:13:04	03APR15:06:05:54	03APR15:01:13:04	03APR15:06:05:54	15	15	4	4	OK
2	WORK	PRODLOC	desc_chg_compare.txt	03APR15:06:22:45	03APR15:06:06:22	03APR15:06:22:45	03APR15:06:06:22	15	15	3	3	Check Files
3	PRODLOC	PRODLOC	desc_chg_l_compare.txt	03APR15:01:19:20	03APR15:06:07:07	03APR15:01:19:20	03APR15:06:07:07	15	15	1	1	OK
4	PRODLOC	PRODLOC	desc_pchg_compare.txt	03APR15:06:06:34	03APR15:01:18:57	03APR15:06:06:34	03APR15:01:18:57	15	15	3	3	Check Files

Display 6. Output dataset after running the macro to check proc compare files

The Proc Compare procedure gives other information such as variable and observation level details. Depending on the requirements, this information can be compared between the two datasets. You can also check through the compare output for notes indicating mismatches. A more detailed account on the details to be checked while looking at a proc compare output can be found in “PROC COMPARE – Worth Another Look!” All these checks can be incorporated for a more detailed check of the files.

CHECKING LOG FILES

Log files need to be checked for errors, warnings or notes of concern. To check for any of these the contents of the log file can be brought into a dataset and then the dataset can be scanned for occurrences of errors, warnings or note phrases. An output dataset can then be created listing the file names and the error/ warning/ note phrases founded in that file. Suzanne Humphreys’s paper “%LOGCHECK: a Convenient Tool for Checking Multiple Log Files” provides a macro that does a fully automated check of log files.

You might want to add some additional checks to the log files. For example, you may want to check if data is being used from the correct location. The best approach with new data transfers is to keep the path for data location, example in the %include file, so that once modified if would apply to all outputs. If you encounter a situation where the path is given within each program the following macro can be used. Macro parameter values that need to be entered are the folder where the log files are located and the log files that you want to check

```
%macro expr (path=, fname=);

%put path entered was &path;
%put file name entered was &fname;

data chk;
  count=put(countw("&fname.", " "), best.);
  call symput ("cnt", count);
run;

%put &cnt;
```

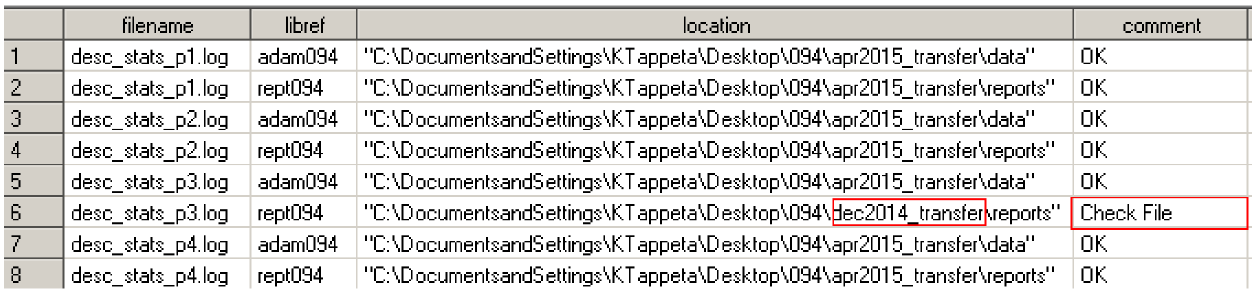
```
%do i=1 %to &cnt;
data param&i.  (keep=libref location filename);
 length line $200. libref $8 location $100;
 retain n;
 infile "&path.%sysfunc(scan(&fname., &i, " "))" length=len;
 input @01 line $varying200. len;
 filename=scan("&fname", &i, " ");…….
 if line ne "";
 if index(strip(line), "libname") > 0;
 libref=scan(line, 3, " ");                           …… (1)
 location=scan(line, 4, " ");
 if location ne "" and libref ne "";
run;

proc append base=param data=param&i.;
run;
%end;

data paramt;
  set param;
  length comment $20;
  if index(location, "apr2015_transfer")>0 then comment="OK";     …… (2)
  else comment="Check File";
  a=3;
run;

%mend expr;
```

(1) Extracting filename, libref and location information from the log file

(2) Checking if all log files have used the April 2015 data

By submitting the below macro, we get the output as shown in display 7. From the output you can see that for one file the libref for the reporting dataset has been assigned to the location that contains the December 2014 transfer instead of the April 2015 transfer.

```
%expr (path=%str(C:\Documents and Settings\KTappeta\Desktop\QC\),
       fname=%str(desc_stats_p1.log desc_stats_p2.log desc_stats_p3.log
desc_stats_p4.log));
```



	filename	libref	location	comment
1	desc_stats_p1.log	adam094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\data"	OK
2	desc_stats_p1.log	rept094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\reports"	OK
3	desc_stats_p2.log	adam094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\data"	OK
4	desc_stats_p2.log	rept094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\reports"	OK
5	desc_stats_p3.log	adam094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\data"	OK
6	desc_stats_p3.log	rept094	"C:\DocumentsandSettings\KTappeta\Desktop\094\dec2014_transfer\reports"	Check File
7	desc_stats_p4.log	adam094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\data"	OK
8	desc_stats_p4.log	rept094	"C:\DocumentsandSettings\KTappeta\Desktop\094\apr2015_transfer\reports"	OK

Display 7. Output Dataset after running macro to check log files

CONCLUSION

Automated checking of multiple files is helpful when there are a large number of files that need to be checked or when changes are made often. Opening each file and checking is time consuming and more prone to error. By bringing required information from these files to a location, it becomes easy to view the contents and identify differences/ errors. The utility macros presented above can be modified depending on the kind of checks required and files being checked. Some functionalities can be added like making the parameter entry more interactive (using %display and %window) in case the user is not familiar with SAS. The results can be viewed as a SAS dataset as shown in this paper or they can be exported to an excel file using proc export depending on the user's preference.

REFERENCES

Humphreys, Suzanne. 2008. "%LOGCHECK: a Convenient Tool for Checking Multiple Log Files." Pharmasug. Vancouver, BC (Canada): Cardiome Pharma Corp.

Williams S., Christianna. 2010. "PROC COMPARE – Worth Another Look!" SAS Global Forum. Durham, NC: Abt Associates Inc.

ACKNOWLEDGMENTS

Percept Pharma Services
1031 US highway
Bridgewater, NJ 08807

Eliassen Group Biometrics and Data Solutions
351, N Frontage road,
New London, CT 06320

CONTACT INFORMATION

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

Name: Kathyayini Tappeta
Enterprise: Percept Pharma Services
E-mail: kathya.tappeta@gmail.com