

**Paper DH15****MACRO TRIO IN SAS – DATASET COMPARISON AND REPORTING****Nagaraju Mancha, Efficacy Analytics Pvt Ltd, Bengaluru, India****ABSTRACT**

SAS is a great tool having macros to automate the repeated tasks. Programmers can develop macros aiding in completing the regular tasks within a short span of time with good quality. Programmers especially validation teams have to validate datasets by comparing validation datasets with production datasets to identify the issues or updates in production datasets. Comparing individual datasets might be easy but if there is a need to compare all datasets at once, definitely it would be a challenge. To overcome this challenge, we can take advantage of macros and automation. This paper illustrates a group of macros which will help programmers to compare all the datasets of libraries at once and create the report which contains mismatches or updates of datasets with clear explanations. The output report of this program will have variable, structural, label and data differences thereby we can improve quality of data and tracing the updates.

This paper illustrates three such amazing macros:

**1. Comparing datasets of two libraries and report creation**

Programmers will be able to compare all datasets of two libraries at once by means of using this macro. In order to execute it successfully programmer have to provide appropriate values to each parameter. Invalid parameter values lead to issue error message in log. PDF and Excel outputs will be generated and placed in the path provided through parameter.

PDF output will have dataset differences with respect to variable, structural, label and data differences. It will also have record count differences of old and new one. Changes are highlighted with color coding so user can easily trace updates. Excel output will have record level differences of each dataset. Programmer can also send these reports to anybody on mail by giving valid value to mail parameter in macro. By means of having this macros updates and issues can easily traced. For unexpected updates and issues, User will have to check with programmer who did provide the datasets being used in this macro.

**2. Compress the datasets ready**

Compress or Zipping the datasets manually might be an easy task however its time consuming and more chances of missing files to be compressed. To overcome these issues This Macro will help in Compressing or Zipping the files and uploading the programmed datasets to any server location - including SharePoint; PINNACLE 21 etc. Programmer will have to provide a valid value to each parameter for success execution.

**3. Create a folder and back-up the datasets.**

Back-up the files and dataset are always good practice for future reference. To know the updates that are occurring during course of time, back-up datasets must be available. While programmers try to execute programs to create the datasets might end up overwriting the existing datasets. Manual process of creating folder and placing the required files into it will take time and have chances to miss out. This macro will help us to create a folder with datetime stamp and place the datasets in the folder for future reference. By doing so we can easily trace the updates or issues between old and current datasets.

## INTRODUCTION

SAS is a great tool having macros to automate the repeated tasks. Programmers having broad knowledge in SAS macros and procedure can take advantage to develop automatic macros aiding in completing the regular tasks within a short span of time with good quality. Most of the organizations will have certain level macros to automate their regular tasks however still manual programming required. Although we do have various methods available to compare datasets, Proc Compare procedure is most proven method to provide differences between two datasets with respect of data, variable, label and format changes.

Clinical Programmers especially validation teams have to validate datasets by comparing validation datasets with production datasets to identify the issues or updates in production datasets. Comparing individual datasets might be easy but if there is a need to compare all datasets at once, definitely it would be a challenge. To overcome this challenge, we can take advantage of macros and automation.

This paper illustrates a group of macros which will help programmers to compare all the datasets of libraries at once and create the report which contains mismatches or updates of datasets with clear explanations. The output report of this program will have variable, structural, label and data differences. Using this program, we can improve the quality of data and tracing the updates.

## 1. COMPARING THE DATASETS OF TWO LIBRARIES AND REPORTING THE CHANGES

### 1.1 WHY DO WE NEED TO COMPARE THE DATASETS?

To ensure the quality of datasets, tracing updates or changes between prior version of datasets and current datasets, to know the difference between the dataset's comparison procedure has to be used.

Double-Programming is always recommended in clinical programming to maintain the quality of outputs in SDTM and AdaM related programming. The validation programmer has to write programs independently as per the standard guidelines and then compare the dataset with dataset created by production programmer.

Programmers often have to run their programs using latest raw datasets to refresh the outputs with latest data. Before executing programs on raw datasets, it is always a good practice to compare earlier raw datasets with latest raw dataset to see the expected data changes.

Proc compare is a primary choice for all programmers to compare the datasets. Syntax and its options as follow

```
proc compare base=sdm.lb compare=vsdm.v_lb out=lb_diff outbase outcomp outnoequal outdiff;  
run;
```

- **Base** = <dataset> and **compare** = <dataset> are datasets to be compared.
- **Out** = <dataset> will be used when we want to see the differences in a dataset.
- **outbase** option writes records from Base dataset
- **outcomp** option writes records from Compare dataset
- **Outnoequal** option Suppress the observations when they are equal in base and compare.
- **outdiff** option writes the observations to output dataset

Proc compare procedure has lot of other options, those can be used as per programmer requirement and please refer SAS documentation (<http://support.sas.com/documentation/>) for complete list of options and its uses.

## 1.2 MACRO 1: %domain\_compare: COMPARISON OF DATASETS

This macro is developed to compare datasets. Macro **%domain\_compare** has following parameters. Programmer must provide appropriate values to each parameter for the program's successful execution. As output report of this program will be saved with date and time stamp, Programmer can execute any number of times in a day.

Programmer will have to take a look at each section of this report for the issues or updates. If there is a potential issue, then get to the bottom of the that by checking programs and data and put these across to the team who has provided the datasets. Proc compare output will not have differences of all records. User can refer the excel file generated by this macro to check the record level mismatches. As the report has highlighted with colors user easily will be able to identify the issues.

If we do get mismatches in report that does not means it has issues. When programs executed on latest data then record count differences are expected so its programmer duty to debug the programs or datasets to check the data to get to know the issue. When there is requirement from AdaM/TLF team to refresh the datasets as latest raw datasets are placed, SDTM programmer will have to execute all programs to reflect the data to SDTM outputs. In lieu of doing, raw dataset of earlier and latest can compare and if there are no changes in raw datasets then no need to re-run SDTM programs. If raw datasets have any issues, those can be reported to Data management team. This program designed for not only SDTM programmers, anybody who wish to compare datasets can use it by providing values to parameters. Just take a look at the below calling macro and its parameters.

```
%domain_compare(prot=TEST-001,
old_path=c:/sdm/datasets_15JAN19,
new_path=c:/sdm/datasets_23JAN19,
output=c:/sdm/compare,
ftype=SDTM, uneqc=Y,
mail_this_report=Y);
```

**Prot=** Protocol number or studyid as it is defined in protocol or SAP. The protocol number will appear in comparison report.

**Old\_path=** Specify the path where earlier version of datasets are available. User must provide correct path to this parameter. It should not have null values. Incorrect or null value of this parameter will issue an error message in log window.

**New\_path=** Specify the path for the latest datasets to be compared with earlier datasets. User must provide correct path else error message will issue in log window.

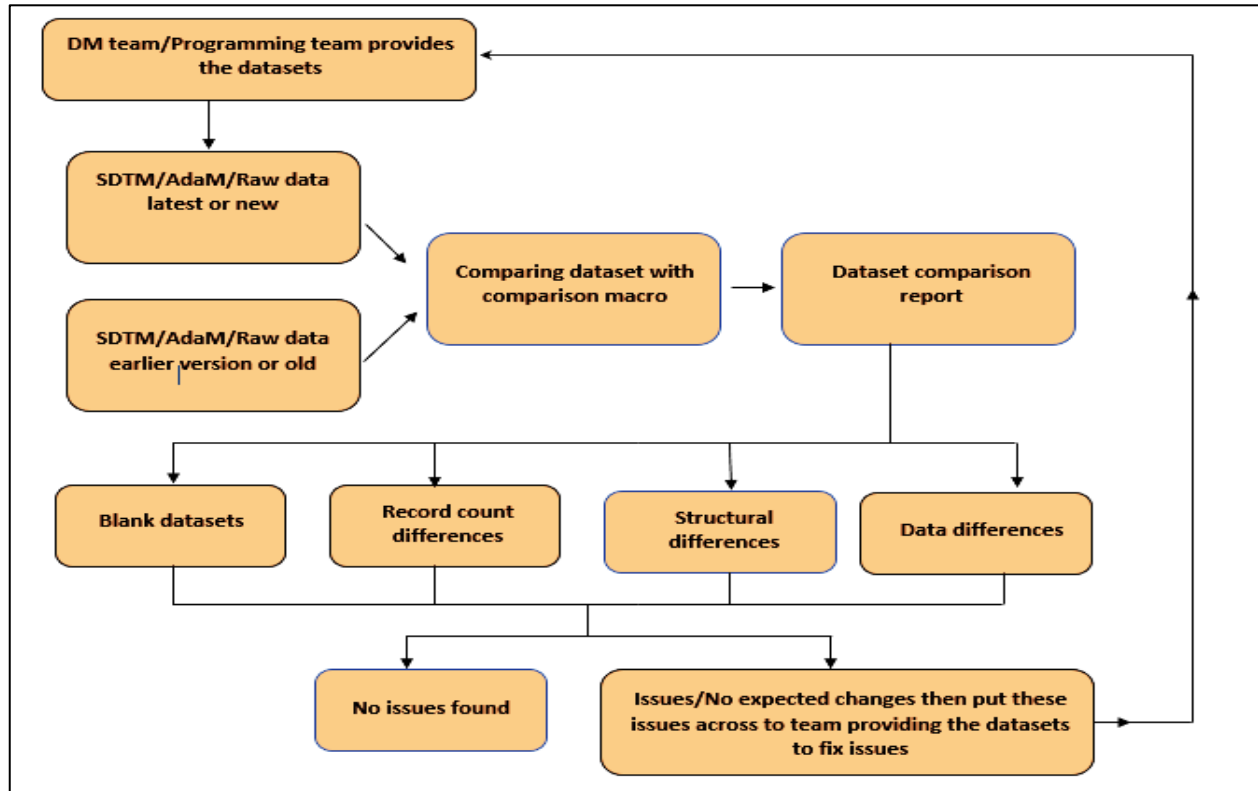
**Output=** Specify the path to save the output comparison report. Comparison report will be saved with date and time at specified location. PDF and excel outputs will be saved in same location specified in the parameter.

**Ftype=** Specify the category of datasets such as SDTM or AdaM. If raw datasets were used, then you can specify as RAW.

**Uneqc=** if two datasets to be compared have unequal records, proc compare output gives lot of discrepancies; so, if we don't want compare datasets having unequal records then specify as N else specify as Y.

**Mail\_this\_report=** if programmer wishes to get this report on mail, specify a value Y to this parameter. Specify N if we don't want this report on mail.(Note: Mail setup feature should be available in SAS installed in your system).

## 1.3 DATASET COMPARISON PROCESS FLOW:



## 1.3 THE OUTPUT - DETAILS OF COMPARISON REPORT:

Comparison report will have programmer name who has created this report, study details, Record count differences, data differences and structure changes such as format changes, variable labels and its type changes.

### STUDY DETAILS:

This section will have programmer name who did generate this report, study name, Path for old and new datasets to be used by compare procedure, List of checks used in macro, path for the old and new datasets being used in macro. Screenshot below for reference:

## STUDY DETAILS

|                                                                                  |                        |
|----------------------------------------------------------------------------------|------------------------|
| STUDY INFORMATION                                                                |                        |
| STUDY:                                                                           | TEST1001               |
| USERNAME:                                                                        | MANCHAN                |
| AUTHOR:                                                                          | EPHICACY ANALYSIS TEAM |
| PROC COMPARE VESION:                                                             | 1.0                    |
| LIST OF CHECKS USED IN THIS PROC COMPARE:                                        |                        |
| 1. RECORD COUNTS BETWEEN OLD AND NEW DATASETS.                                   |                        |
| 2. EXISTENCE OF DATASETS BETWEEN TWO LIBRARIES.                                  |                        |
| 3. FINDING THE VARIABLE DIFFERENCES IN LENGTH,TYPE,FORMAT, LABEL AND ITS VALUES. |                        |
| 4. COMPARING OLD DATASETS WITH NEW DATASETS AND EXPORTING THE DIFFERENCES        |                        |
| PATH FOR THE OLD SDTM DATASETS: C:/SDTM/OLD                                      |                        |
| PATH FOR THE NEW SDTM DATASETS: C:/SDTM/NEW                                      |                        |

### 1.4 DATASETS DOES NOT HAVE DATA OR BLANK DATASETS:

Sometimes datasets might not have a data to compare. Programmer can see the list of blank or null datasets. Null datasets are caused by programming issue or data may not be available in database to map. Programmer has to check the respective program or the raw datasets that are being used to create output. If null datasets are caused by programming issues, then programmer has to fix the issue to get the data to output. If data is not available, then user has to take this data issue to the team providing the datasets.

## BLANK DATASETS WITH NO OBSERVATIONS

| Comments                                                  | library | Dataset |
|-----------------------------------------------------------|---------|---------|
| These domains don,t have data please check and confirm... | NEW     | IE      |
| These domains don,t have data please check and confirm... | NEW     | SUPPIE  |

For example - as per the above report IE domain does not have data. This is an expected issue since if there are no screen failure subjects then IE domain will not have data. If DM domain has screen failure subjects but those are not mapped to IE domain then it is an issue with IE domain programming.

### 1.5 RECORD COUNT DIFFERENCES BETWEEN DATASETS:

There is a chance to have unequal number of records between the old and latest datasets. Data might have been included or deleted in current datasets. Following report will help us to know the datasets having unequal observations. In this report green color highlights newly added records and red color highlights deleted records. In the below report MH and RELREC are highlighted with red color as the data has been deleted. Programmer has to check the corresponding program for any issues which causes deletion or addition of records.

## RECORD COUNT COMPARE OF OLD DATASET VS NEW DATASET

| Dataset Library Name | Dataset Name | Records in Old Library | Old Library dataset creation date | Records in New Library | New Library dataset creation date | Comments                       |
|----------------------|--------------|------------------------|-----------------------------------|------------------------|-----------------------------------|--------------------------------|
| NEW                  | AE           | 230                    | 21AUG18:15:52:35                  | 242                    | 23JAN19:12:55:50                  | 12 Records have been Included  |
| NEW                  | CM           | 437                    | 21AUG18:15:52:46                  | 445                    | 23JAN19:12:56:10                  | 8 Records have been Included   |
| NEW                  | DA           | 706                    | 21AUG18:15:52:58                  | 786                    | 23JAN19:12:56:21                  | 80 Records have been Included  |
| NEW                  | DM           | 31                     | 21AUG18:15:54:28                  | 31                     | 23JAN19:12:59:26                  |                                |
| NEW                  | DS           | 49                     | 21AUG18:15:53:15                  | 52                     | 23JAN19:12:56:26                  | 3 Records have been Included   |
| NEW                  | EG           | 138                    | 21AUG18:15:53:19                  | 138                    | 23JAN19:12:56:33                  |                                |
| NEW                  | EX           | 42                     | 21AUG18:15:53:36                  | 42                     | 23JAN19:12:56:38                  |                                |
| NEW                  | IE           | 0                      | 21AUG18:15:52:39                  | 0                      | 23JAN19:12:55:55                  |                                |
| NEW                  | LB           | 5733                   | 14AUG18:12:40:24                  | 6161                   | 23JAN19:12:58:10                  | 428 Records have been Included |
| NEW                  | MH           | 449                    | 21AUG18:15:53:40                  | 401                    | 23JAN19:12:56:45                  | 48 Records have been Deleted   |
| NEW                  | MI           | 63                     | 21AUG18:15:54:11                  | 65                     | 23JAN19:12:57:10                  | 2 Records have been Included   |
| NEW                  | MQ           | 189                    | 21AUG18:15:52:51                  | 204                    | 23JAN19:12:56:15                  | 15 Records have been Included  |
| NEW                  | PR           | 130                    | 21AUG18:15:53:46                  | 136                    | 23JAN19:12:56:52                  | 6 Records have been Included   |
| NEW                  | QS           | 540                    | 21AUG18:15:53:55                  | 555                    | 23JAN19:12:56:58                  | 15 Records have been Included  |
| NEW                  | RELREC       | 307                    | 31JUL18:20:03:27                  | 268                    | 23JAN19:12:59:21                  | 39 Records have been Deleted   |

### 1.6 VARIABLE LEVEL CHANGES:

Variable level changes such as inclusion or exclusion of variables, changing in variable labels, formats and its length can be seen in the below report. Adding new dataset or deleting existing dataset also is reported in this report. Programmer has to go through this report to know the variable level changes across all datasets in old and new libraries.

## VARIABLE LEVEL REPORT OF OLD DATASET VS NEW DATASET

| Obs | domain  | name     | old | new | comment                                                      |
|-----|---------|----------|-----|-----|--------------------------------------------------------------|
| 1   | DV      |          |     |     | The domain DV has been added please check                    |
| 2   | LB_2709 |          |     |     | The domain LB_2709 has been added please check               |
| 3   | SUPPDV  |          |     |     | The domain SUPPDV has been added please check                |
| 4   | SUPPLB  |          |     |     | The domain SUPPLB has been added please check                |
| 5   | LB      | LBSCAT   |     |     | The variable LBSCAT has been added in LB..Please check..     |
| 6   | MI      | MIREASND |     |     | The variable MIREASND has been deleted in MI..Please check.. |

### 1.7 DATA DIFFERENCE BETWEEN DATASETS:

This part will have proc compare report of each dataset used in the macro. So, programmer can see data difference between earlier and current datasets. Each output contains dataset summary, variable 7

summary and observation summary. Compare report gives the differences in datasets having unequal observations so this macro has a parameter UNEQC=. If we want to run compare procedure on datasets



even having unequal no of observations, then use UNEQC=Y in calling macro. Else change the parameter value to N. Below report showcases this:

**Bookmarks**

- VARIABLE LEVEL REPORT OF OLD DATASET VS NEW DATASET
  - comparing old.DM with new.DM
  - comparing old.EG with new.EG
  - comparing old.EX with new.EX
  - comparing old.IE with new.IE
  - comparing old.SUPPDM with new.SUPPDM
  - comparing old.SUPPDS with new.SUPPDS
  - comparing old.SUPPEG with new.SUPPEG
  - comparing old.SUPPIE with new.SUPPIE
  - comparing old.SUPPPR with new.SUPPPR
  - comparing old.TA with new.TA
  - comparing old.TE with new.TE

02:25 Monday, January 28, 2019 9

**SDTM COMPARE REPORT FOR THE STUDY:TEST1001**

**COMPARING EX(21AUG18:15:53:36) WITH EX(23JAN19:12:56:38)**

The COMPARE Procedure  
Comparison of OLD.EX with NEW.EX  
(Method=EXACT)

Data Set Summary

| Dataset | Created          | Modified         | NVar | NObs | Label    |
|---------|------------------|------------------|------|------|----------|
| OLD.EX  | 21AUG18:06:23:36 | 21AUG18:06:23:36 | 18   | 42   | Exposure |
| NEW.EX  | 23JAN19:02:26:38 | 23JAN19:02:26:38 | 18   | 42   | Exposure |

Variables Summary

Number of Variables in Common: 18.  
Number of Variables with Differing Attributes: 1.

Listing of Common Variables with Differing Attributes

| Variable | Dataset | Type | Length | Label                      |
|----------|---------|------|--------|----------------------------|
| EXADJ    | OLD.EX  | Char | 33     | Reason for Dose Adjustment |
|          | NEW.EX  | Char | 36     | Reason for Dose Adjustment |

Observation Summary

This macro will also create an excel output where we can see the list of changes in each dataset. Proc compare output and excel output will be placed at the path defined in parameter output=<Path>. Each dataset differences will be representing in each sheet in same excel file. Below report screenshot for reference. We can get these reports to our mail by enabling the parameter 'mail\_this\_report=Y'.

SDTM\_compare\_23JAN2019\_with\_21AUG2018\_2143.xlsx - Microsoft Excel

| I      | J       | K      | L       | M       | N       | O                                   | P        | Q                | R            | S      | T      | U |
|--------|---------|--------|---------|---------|---------|-------------------------------------|----------|------------------|--------------|--------|--------|---|
| EXDOSE | EXDOSTX | EXDOSL | EXDOSFR | EXDOSFR | EXROUTE | EXADJ                               | EPOCH    | EXSTDTC          | EXENDTDC     | EXSTDY | EXENDY |   |
| 80     |         | mg     | CAPSULE | QD      | ORAL    | RE-START DOSING                     | TREATMEN | 2017-02-02T18:10 | 2017-11-07   | 221    | 499    |   |
| 80     |         | mg     | CAPSULE | QD      | ORAL    | OTHER:PT FORGOT TO DOSE ON 01FEB201 | TREATMEN | 2017-02-02T18:10 | 2017-11-07   | 221    | 499    |   |
|        |         |        |         |         |         | XXXXXXXXXX.XXXXXX.XXXXXX.XXXXXXXX   |          |                  |              |        |        |   |
| 160    |         | mg     | CAPSULE | QD      | ORAL    | NOT APPLICABLE, FIRST DOSE          | TREATMEN | 2016-07-11T19:35 |              |        | 1      |   |
| 160    |         | mg     | CAPSULE | QD      | ORAL    | NOT APPLICABLE, FIRST DOSE          | TREATMEN | 2016-07-11T19:35 | 2018-11-27   | 1      | 870    |   |
|        |         |        |         |         |         |                                     |          |                  | XXXXXXXXXXXX |        |        |   |
| 160    |         | mg     | CAPSULE | QD      | ORAL    | NOT APPLICABLE, FIRST DOSE          | TREATMEN | 2016-08-15T07:15 |              |        | 1      |   |
| 160    |         | mg     | CAPSULE | QD      | ORAL    | NOT APPLICABLE, FIRST DOSE          | TREATMEN | 2016-08-15T07:15 | 2018-12-31   | 1      | 869    |   |
|        |         |        |         |         |         |                                     |          |                  | XXXXXXXXXXXX |        |        |   |

## 1.8 REPORTS ON e-mail:

We can get the outputs of this macro to our mail by enabling the parameter 'mail\_this\_report=Y'. Assigning N to the parameter mail\_this\_report will not provide the reports to the mail. The following piece of code has been included in macro to get the report to user mail. If programmer wishes to send this report to other team members, other email id's can also be included in the program.

```
%if %upcase(&mail_this_report)= Y %then %do;

FILENAME myemail EMAIL from="nagaraju.mancha@ephicacy.com"
to=( "nagaraju.mancha@ephicacy.com" )
cc="nagaraju.mancha@ephicacy.com"
Subject = "Comparing Report: ftype._compare_&new._with_&old._%sysfunc(compress(&tm,','))"
type='text/html';

data _null_;
file myemail;
put "Datasets created on &old has been compared with Datasets created on &new"
/
"Comparison outputs: &ftype._compare_&new._with_&old._%sysfunc(compress(&tm,','))pdf and
&ftype._compare_&new._with_&old._%sysfunc(compress(&tm,','))xlsx has been placed at &output";

put "Thanks & Regards,";
put "&sysuserid"
// " ";
run;

%end;
```

## 1. MACRO 2: %create\_zip: COMPRESSING DATASETS PROGRAMATICALLY:

Sometimes we need to compress or zip the files to effectively post them on any document repository or place them in standard web tools such as pinnacle 21 to validate the datasets. Files can get corrupted and might consume time when they are transferred over the mail. So, file compression is necessary to save space and to reduce time for transmission over mail. It is easier to compress multiple files to one file and attach to mail or upload them on web tool rather than uploading individual file.

The following macro is designed to compress files to upload or send over the mail. The macro has following parameters; user has to provide appropriate values to compress the files. Folder to be compressed can have multiple types of files, it is possible to compress selectively specific types of files by specifying file extension as a parameter value. Compressed folder name would be same name as folder name. Programmer often have to compress files so This is easiest method to compress or zip the files Despite multiple tools available on internet.

```
%create_zip(ftype=xpt, path=/sdtm/xpt);
```

**Ftype=** file type that we need to select to compress. Folder can have multiple types of files such as xpt, xls, xlsx, doc, etc. if you want compress selectively specific type of file then we have to give file type to this parameter. If you don't specify macro will compress all the files available in folder.

**Path=** A valid path where files available to be compressed. Wrong path leads to an issue error message in the log window. Zip or compressed file will be saved in the path specified in the parameter path.

The following program will help you to understand what procedures and steps has been used to write this macro. ODS PACKAGE has been used inside the macro. This macro will work in Linux and windows environment. By means of having this macro we can reduce issues significantly and avoid the issues.

## PROGRAM TO COMPRESS/ZIP THE FILES:

```
%macro create_zip(ftype=,path=);
%local dsnpath path dsnpath;

data getfiles;
keep filename;
length fref $8 filename $80;
rc=filename(fref,"&path");
if rc =0 then
did=dopen(fref);
else if rc < 0 then
putlog 'ERR' 'OR: Unable to open directory.';
dnum=dnum(did);

do i= 1 to dnum;
filename=dread(did,i);
fid=mopen(did,filename);

if fid > 0 then
output;
end;
rc = dclose(did);
run;

proc sql;
create table sdtmeta as select * from getfiles
where upcase(filename) like "%quote(%).%upcase(&ftype)";
select lowercase(filename) into: tmp1 - :tmp&sqllobs from sdtmeta;
quit;

%let dsnpath=&path;
ods package (sdtmoutput) open nopf;

%do i= 1 %to &sqllobs;
%put &dsnpath./&&tmp&i;

%if %sysfunc(fileexist("&dsnpath./&&tmp&i")) %then
%do;
%put file: &&tmp&i...xpt exist;
ods package (sdtmoutput) add file="&dsnpath./&&tmp&i";
%end;
%else
%do;
%put WARNING: XPT file &&tmp&i not available please check;
%end;
%end;

ods package (sdtmoutput) publish archive
properties (archive_name="datasets_&sysdate..zip" archive_path="&dsnpath");
ods package (sdtmoutput) close;
%mend;

%create_zip(ftype=xpt,path=C:/sdtm/xpt);
```

## 2. Macro 3 - %files\_backup: CREATE FOLDER AND BACK-UP THE DATASETS

Back-up is performed by copying files to other location for future reference and tracing the updates. It is very important to do regular backups to prevent loss of data. Manually creating folder and placing the data into folder is time consuming activity and this process may end up with error. We can write a macro to create folder with a name date and placing datasets into folder programmatically. The following macro will help us to do this task. Programmer has to provide appropriate parameters in order to execute.

**%files\_backup**(copyin=,copyout=,ftype=rtf);

**Copyin=** Provide the path where the datasets or files are available to be copied.

**Copyout=** Provide a path where copied datasets have to be placed. If you want to create a new folder, folder name can be given in the same path. Macro will create the folder and place files into it.

**Ftype=** Folder can have multiple types of file or datasets. If we want to copy certain types of file, then we have to provide file extension so that macro will select all files having the same extension and place in a desired destination. If we don't provide file extension, macro will copy all files and place them in the path provided in copy out parameter.

### SAMPLE PROGRAM:

```
%macro files_backup(copyin=,copyout=,ftype=);
%LOCAL rc fileref return;
%let rc = %sysfunc(filename(fileref,&copyout));

%if %sysfunc(fexist(&fileref)) %then
    %let return=1;
%else %let return=0;
%put &return;

%if &return=0 %then
    %do;
        x mkdir "&copyout";
        %put folder has been created successfully...;
        x "cp &copyin/*.&ftype &copyout";
    %end;

%if &return > 0 %then
    %do;
        %put Warning: Folder has existed with same name..;
    %end;
%mend files_backup;
```

### CONCLUSION:

Macros are a boon for programmers. In this paper, the three macros provide an advantage edge for programmers. Dataset comparison will let us know the difference between datasets. This is a best practice to compare datasets. We can reduce the timelines and improve the quality of datasets leading to delivering quality data to client or submitting the data to regulatory agencies. This will help us optimize the process and reduce manual intervention significantly. Clinical programmers, especially working on SDTM, AdaM and TLF programming can use these macros to get quality outputs. Clinical programmers will get the EDC data on regular basis and they have to run the programs regularly to reflect the fresh data into outputs. Using the macros, it will be easy to know the updates by comparing earlier datasets with current datasets. Back up of the datasets also can be done using the program defined here. The goal is to aid the programming community with automated weapons in the form of macros, which will help us fight the dreaded world of diseases more efficiently.



## REFERENCES:

- Proc compare procedure and its options Retrieved from SAS Documentation  
<https://documentation.sas.com/?docsetId=proc&docsetTarget=n0c1y14wyd3u7yn1dmfcpaejllsn.htm&docsetVersion=9.4&locale=en>
- Macro functions used in this paper referred from SAS support  
<https://documentation.sas.com/?docsetId=mcrolref&docsetTarget=titlepage.htm&docsetVersion=9.4&locale=en>
- SAS macro programming made easy by Michele M. Burlew
- Carpenter's Complete guide to the SAS macro language.

## ACKNOWLEDGEMENTS:

I would wish to express my thanks to Efficacy colleagues and management involved in providing inputs and reviewing the content.

## CONTACT INFORMATION:

Your queries, comments and questions are always valued and encouraged. You can reach out me at:

Nagaraju Mancha

Efficacy Analytics Pvt Ltd

[Nagaraju.mancha@efficacy.com](mailto:Nagaraju.mancha@efficacy.com)

[Nagaraju.mancha@yahoo.com](mailto:Nagaraju.mancha@yahoo.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.