Paper SM13

Streamlining the Integration of SDTM Datasets from Multiple Studies: A Macro Approach

Santosh Reddy Lekkala, Medidata, Morrisville, USA
Vijay Anuroop Reddy Allam, Medidata, Morrisville, USA

ABSTRACT

In New Drug Application (NDA) or Biologics License Application (BLA) submissions to the FDA, Integrated Summary of Safety (ISS) and Integrated Summary of Efficacy (ISE) play pivotal roles. Building integrated datasets by pooling data from multiple studies is a complex and time-consuming task which is either done at Study Data Tabulation Model (SDTM) or Analysis Data Model (ADaM) stage. This PHUSE paper presents a macro designed to streamline the data integration process, particularly focusing on SDTM datasets. We will address common challenges encountered during dataset integration, such as variables with different lengths, missing data and variable label discrepancies. The macro will facilitate the identification of these inconsistencies and generate an output to help resolve them effectively thereby saving valuable time and effort. By automating these intricate steps, the macro significantly enhances data quality, reduces time, and brings in better consistency, thereby improving the likelihood of a successful submission.

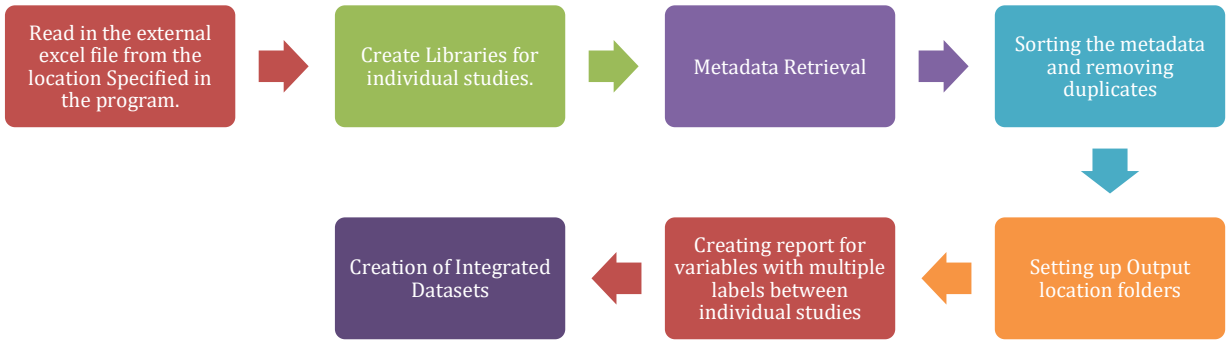
INTRODUCTION

The following SAS macro is a comprehensive tool to streamline the process of integrating data from multiple studies for Summary of Safety (ISS), Integrated Summary of Efficacy (ISE) or for analyzing data coming from multiple studies enables the harmonization of diverse datasets, ensuring a unified and organized output dataset. Though the current macro focuses on integrating SDTM datasets, it can be used to stack ADaM or raw datasets with the same dataset names for any number of studies.

Advantages:

1) The macro not only performs integration but also serves as an edit check tool when combining SDTM/ ADaM datasets from multiple sources. It generates a report for variables having different labels across studies so that they can be investigated and updated as required.
2) It saves a considerable amount of time, as it auto-adjusts the length of the same variables with different lengths across multiple datasets.
3) The macro can lessen the workload for Programmers by automating tasks and improving data quality, ultimately leading to cost savings

Process Flow:



Read in the excel file:

The Input for the Macro is an excel file that contains the global macro variables that would be used in the program. This file contains 3 main input columns – Type, StudyID and Location.

Type: The Type column is used to identify if the row relates to Input or Output location.

StudyID: Used to specify the StudyID associated with the location. For this macro the length of the StudyID cannot be more than 7. StudyID column would be blank when type = “Output Location”.

Location: Specifies the location where the datasets to be integrated are available.

The PROC IMPORT step reads the Excel file into a SAS dataset.

Screenshot 1: Input Excel file columns:

Type	StudyID	Location
Output Location		X:\Path\Output
Input Location	1	X:\Study datasets path
Input Location	2	X:\Study datasets path
Input Location	3	X:\Study datasets path
Input Location	4	X:\Study datasets path

Creating libraries for individual studies:

Following the PROC IMPORT of input data, the macro utilizes SQL queries to extract essential information from the excel sheet and uses SQL queries serve multiple purposes such as counting the unique number of Studyid’s (&N), Extracting StudyId’s (&LIB1 - &LIB&N.) and extracting corresponding data locations (&PATH1 - &PATH&N.). This information is used to create libraries for individual studies specified in the Excel sheet.

Metadata retrieval:

The macro retrieves metadata for datasets in each study library. This is achieved using the PROC DATASETS procedure, which provides a detailed overview of the structure and attributes of datasets within the specified libraries. The metadata from all the studies is combined in a single dataset and is used in subsequent processing steps.

Sorting the metadata and removing the duplicates:

The combined metadata obtained from all the studies in the above step is sorted by domain, variable, and Max length for each variable. Duplicates are identified and removed, retaining the maximum variable length. The resulting dataset serves as a clean representation of the metadata. The dataset is then sorted by domain and varnum so that the order of variables remains similar. The resulting dataset contains attributes to include the variable name, label, length, and data type. The macro utilizes the CALL EXECUTE routine to dynamically generate data step code for creating individual SDTM data shells based on the metadata. These shells would be later used to append the SDTM datasets from individual studies.

Setting up output location folder:

To set up the output folder the Macro reads in the location from the Excel file where TYPE = “Output Location” and the program run date is used to create a folder in the provided location. The folder would be named in YYYYMMDD format based on the date. Following this step, 3 other subfolders are created in the dated folder using the LIBNAME statements – INTERIM: to store intermediate datasets, INTDATA – location to store the final integrated datasets, INTCSV – location to save the integrated CSV files.

Creating a report for the same variables with multiple labels across studies:

In the step above – “Sorting the metadata and removing the duplicates” when sorted by domain and variable names either in SDTM or ADAM we see that there might be differences in the labels used from one study to another. This might be caused by several factors such as different I.G versions being used, label errors, etc., and needing to be corrected. A report is generated for variables with multiple labels using the PROC EXPORT to generate an Excel file. The resulting Excel file, named "Variables with Multiple Labels.xlsx," provides the list of domains and variable labels that needs to be investigated.

Creation of Integrated Datasets:

In the final step the datasets from individual studies are got into the work library with “_studyid” added as a suffix. The datasets are then set with the shells that were created earlier to get the final Integrated datasets and CSV files.

Integration Macro Code:

```
/***Delete existing datasets in work library****/
PROC DATASETS LIB=WORK KILL NOLIST;
QUIT;

************************** MACRO INPUTS ************************;
%global file;
* Update file path for the excel file which contains the input;
%let file=file path/file name.xlsx;
PROC IMPORT DATAFILE="&FILE." OUT=excel DBMS=XLSX REPLACE ;
 GETNAMES=YES;
RUN;

data excel_ ;
 set excel ;
  if type = "Input Location" ;
 _studyid = "_"||strip(put(studyid,best.)) ;
run ;

PROC SQL NOPRINT;
   SELECT COUNT(StudyID) INTO :N
   FROM excel_;
   SELECT _StudyID INTO  :LIB1 - :LIB%LEFT(&N.)
   FROM excel_;
   SELECT Location INTO :PATH1 - :PATH%LEFT(&N.)
   FROM excel_;
QUIT;

%LET L = %LEFT(&N);
%PUT &L.;

/* CREATE THE LIBRARY FOLDERS*/

options validvarname=any;
%macro lib;
%DO I = 1 %TO &L.;
   LIBNAME &&LIB&I.. "&&PATH&I.." access=readonly;
%END;
%mend;

%lib;

%macro meta ;

%DO I = 1 %TO &L.;
 PROC DATASETS LIBRARY=&&LIB&I.. MEMTYPE=DATA NOPRINT;
   CONTENTS DATA=_ALL_ OUT=WORK.&&LIB&I..;
 QUIT;
%END;

 data c1 ;
  set _: ;
 run ;
%mend ;
%meta;



* Sorting the metadata from all the studies by domain, variable and max length for
each variable ;

proc sort data = c1 ;
 by memname name descending length ;
run ;
```

```
* Duplicates in Label ;

proc sort data = c1 nodupkey out=dup_1 ;
 by memname name label ;
run ;

data dup_2 ;
 set dup_1 ;
 by memname name ;
 if first.name then dup = 1 ;
 else dup + 1 ;
 if dup ne 1 ;
 keep memname name ;
run ;

proc sort data = dup_2 nodupkey out=dup_3; by memname name ; run ;

data dup_4 ;
 merge c1(in=a) dup_3(in=b) ;
 by memname name ;
 if a and b ;
 keep libname memname name label ;
run ;


* Removing duplicates and taking the maximum length available for a variable ;
proc sort data = c1 nodupkey out=c2 ;
 by memname name ;
run ;

* Sorting the dataset by domian and varnum ;
proc sort data = c2 out=c3 ;
 by memname varnum ;
run ;

/* Generate Empty Datasets */

 data c4 ;
 set c3 ;
 by MEMNAME ;
 if type= 2 then type1 = "text" ;
 else if type = 1 then type1 = "num" ;

 if first.MEMNAME then call execute("data
work."||strip(MEMNAME)||"(label='"||strip(MEMLABEL)||"');");
 if first.MEMNAME then call execute("attrib ");

 call execute(""||strip(NAME)||" ");
 call execute("label='"||strip(Label)||"' ");

 if Type1="text" then call execute("length=$"||strip(put(Length,best.))||" ");
 else call execute("length="||strip(put(Length,best.))||" ");

 if last.MEMNAME then call execute(";call missing(of _all_);run;");
 if last.MEMNAME then call execute("proc sql; delete * from
work."||strip(MEMNAME)||";quit;");
run;


* Creating folder and assigning output location;
* To get the date ;

data _null_;
 dt=compress(tranwrd(put( today( ),is8601da. ),'-',''));
 call symput('dt',strip(dt));
run;
```

```
%put &dt;

data excel_a1 ;
 set excel ;
  if type = "Output Location" ;
  call symput('outpath',strip(location)) ;
run ;

%put &outpath;

options dlcreatedir;
libname folder "&outpath.\&dt.";

%let inter=%trim(&outpath.\&dt.);
%put &inter;

options dlcreatedir;
libname intrim "&inter.\Intermediate Datasets";
libname intdata "&inter.\Integrated datasets" ;
libname intcsv "&inter.\Integrated csv" ;


PROC EXPORT
DATA=dup_4
DBMS= xlsx Replace
OUTFILE="&inter\Variables with Multiple Labels.xlsx";
run;

* Setting all Datasets together;

/* 1. getting all datsets into on elibrary with suffix studyid*/

%macro rename_var(inds=, outds=, study=);

 PROC IMPORT DATAFILE="&FILE." OUT=excel DBMS=XLSX REPLACE ;
  GETNAMES=YES;
 RUN;

data excel;
 set excel;
 where STUDYID ne .;
run;

libname outds "&outds";

        data null;
         set excel  end=eof;

         call symputx(strip('_studyid'||strip(put(_n_,best.))),(studyid));
       call
symputx(strip('_location'||strip(put(_n_,best.))),upcase(strip(location)));
                if eof then
                        call symputx('_xlnum',_n_);

        run;

       %put &_studyid1;
       %put &_location1;
       proc sql ;
select count(distinct studyid) into: totcount
from null;
quit;

%put &totcount.;
```

```
        %do i=1 %to &totcount.;

       %put i.;

                %put <<<Study name: &&_studyid&i >>>;
                    %put <<<Loacation: &&_location&i >>>;
                 libname inds "&&_Location&i";
                 proc copy in=inds out =outds memtype=data;
                     run;



      %put &_studyid1;
      %put &_location1;


   proc sql;
   create table work.control as
   select memname, (strip(memname)|| "_&&_studyid&i") as newname
   from dictionary.tables
   where libname='OUTDS' and index (memname,"_")=0;

quit;


data _null_;
   set work.control end=lastone;
   if _n_=1 then call execute('proc datasets library=outds;');
   call execute ('change '||memname||'='||newname||';');
   if lastone then call execute('quit;');
run;

 %end;
 %mend rename_var;

%rename_var (outds=%str(&inter.\Intermediate Datasets));

 data c4_1 ;
  set c3 ;
  by MEMNAME ;

  if first.MEMNAME then call execute("data
intdata."||strip(MEMNAME)||"(label='"||strip(MEMLABEL)||"');");
  if first.MEMNAME then call execute("set work."||strip(MEMNAME)||"
outds."||strip(MEMNAME)||"_: ;");

 if last.MEMNAME then call execute("run;");
 if last.MEMNAME then call execute("proc sql; delete * from
work."||strip(MEMNAME)||";quit;");
run;
```

CONCLUSION

In summary, the provided SAS macro is a comprehensive tool for data integration, metadata management, and dataset organization. It is designed to handle different datasets from multiple studies, ensuring a standardized output. The macro is suitable for various scenarios and data structures such as SDTM, ADaM, or Adhoc analysis where dataset names/structures are similar.

ACKNOWLEDGMENTS

Thanks to the Medidata colleagues and management for their support in making this happen and a special thanks to Bhargav Koduru for his time and valuable feedback during the writing of this paper.

CONTACT INFORMATION

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

Author Name: Santosh Reddy Lekkala
Company: Medidata
Address: Morrisville, NC
Email: lsantosh280@gmail.com

Author Name: Vijay Anuroop Reddy Allam
Company: Medidata
Address: Morrisville, NC
Email: vijayanuroop@gmail.com