SESUG Paper 58-2025

Ensure complete raw data to SDTM mapping: a SAS macro approach for quality control

Cally Davidson, Eric Howard, Rho, Inc.

ABSTRACT

Accurate and complete mapping of raw clinical data to Study Data Tabulation Model (SDTM) domains is essential for high-quality regulatory submissions. A common oversight occurs when raw datasets contain observations that appear only later in the study, such as pregnancy data within the RP domain, leading to incomplete or missing SDTM domains if initial mapping was based on earlier data snapshots. This paper highlights a SAS macro, called search_raw, developed to automate the quality control (QC) process by ensuring all relevant raw data sources with observed records are mapped to SDTM domains. The macro checks for the presence of data from raw sources in the .SAS logs and summarizes SDTM input source datasets, while flagging any unmapped data streams. The RP domain serves as a case study, illustrating how subjects with no pregnancies reported early in the study can be missed if mappings are not revisited. By automating these checks, the macro enhances traceability, reduces manual review, and ensures complete and compliant SDTM deliverables.

INTRODUCTION

In clinical research, the transformation of raw study data into standardized formats is crucial for regulatory compliance and efficient data review. The Study Data Tabulation Model (SDTM), established by the Clinical Data Interchange Standards Consortium (CDISC), serves as a foundational standard for submitting clinical trial data to regulatory agencies such as the FDA. However, accurately mapping raw data to SDTM domains presents ongoing challenges especially when clinical observations emerge later in the study. For instance, planned domains like Reproductive System Findings (RP) may initially appear to contain no relevant data in early stages of the study, only for later in the study events might become reported. When initial data snapshots are used to complete SDTM programming without ongoing review, essential subject data may be omitted, potentially compromising submission quality. This paper presents an automated SAS macro, search_raw, designed to streamline the quality control (QC) process for SDTM mapping, ensuring completeness and regulatory compliance.

AUTOMATING QC WITH THE SEARCH_RAW SAS MACRO

The search_raw macro addresses a common oversight in clinical data programming: failing to revisit raw data sources throughout the duration of the study. Early mapping decisions based on incomplete datasets often miss important records that appear later in the study. Most study teams do not spend time programming a domain until there is data present, since that domain would not be included in the submission packages if there are no observations. The macro is helpful to be run after all SDTM domains programming and spec'ing have been completed, ideally before a draft delivery and again before a final delivery. The macro expects all SDTM programs to be batched, so that it can search the latest logs. This functionality not only improves traceability between raw and SDTM datasets but also reduces reliance on manual review processes, which are prone to error and often lack consistency. The macro ensures such critical information is captured by continuously validating that all applicable raw data streams are considered in SDTM development.

REPRODUCTIVE SYSTEM FINDINGS (RP) CASE STUDY

Study teams typically only program a domain once data is available. Many times, the RP domain is only expected if a pregnancy is reported on the CRF. In this scenario, a study was ongoing and had a draft delivery, however at the data cut for the draft there were no pregnancies in RAW.PGFUMSTR (Pregnancy Follow-up). The team then received a new data transfer closer to database lock and reran Search_Raw macro. There were now observations and the PGFUMSTR row was highlighted in the output file. Consistently running this macro at the time of SDTM creation for a draft and again before database lock alerted the team of potential missed data streams that otherwise may have been forgotten.

MACRO PARAMETERS AND DEFINITION

The following is the macro with parameters used in the search_raw macro:

```
%search_raw(RAWdir  =, SDTMdir =, outPath =, filenam =);
```

- RAWDir: location of where the raw dataset live
- SDTMDir: location of where the SDTM program .log files
- outPath: location of where the results should be outputted
- filenam: filename of the outputted Excel file

Example macro call:

```
%search_raw(RAWdir  = S:\StudyABC\Biostatistics\Data\Clinical,
            SDTMdir = S:\StudyABC\Biostatistics\Prog\SDTM,
            outPath = S:\StudyABC\Biostatistics\Prog\SDTM\QC,
            filenam = Raw data check_&sysdate.);
```

The macro is designed to evaluate the use of raw datasets in SDTM programming by combining information from the raw data directory and corresponding SDTM program logs. When executed, it first creates an internal dataset listing all .sas7bdat files located in the ‘RAWdir’ folder. PIPE options for both Windows and Unix operating environments are used. For each of these raw datasets, the macro calculates the number of observations, storing these counts for later inclusion in the final output.

```
   filename PIPED pipe %if &sysscp.=WIN %then "dir /b ""&RAWdir"" ";
   %else "ls ""&RAWdir"" ";;
      data dtfiles;
         infile PIPED pad;
         input datasets $256.;
         raw_ds = upcase(scan(datasets,1,"."));
         call symput('nbfiles','1');
         *Selecting only sas7bdat files, no folders;
         if index(datasets,".sas7bdat");
      run;
   filename PIPED clear;

   libname raw "&RAWdir" access=readonly inencoding=any;
   proc sql;
      create table DS_obs as
      select upcase(memname) as raw_ds, nobs
      from dictionary.tables
      where libname='RAW'
      ;
   run;
```

To identify which raw datasets are used in SDTM programming, the macro employs the ‘PIPE’ function to search the 'SDTMdir’ folder for all associated SDTM .log files. If no log files are found, a warning is generated indicating that no files were located in the specified directory.

```
   filename PIPED pipe %if &sysscp.=WIN %then "dir /b ""&SDTMdir"" ";
   %else "ls ""&SDTMdir"" ";;
      data PROGS;
         infile PIPED pad;
         input sdtmprog $256.;
         if index(sdtmprog,".log");
```

```
        call symput('nbfiles','1');
    run;
filename PIPED clear;
```

When log files are present, the macro parses each file to locate lines in the .log, containing both the key phrases “NOTE: THERE WERE” and “DATA SET RAW.”. Only lines containing both key phrases are retained, to obtain which raw datasets were utilized in the SDTM creation. Note that this search assumes the raw data library reference is RAW; if a different LIBNAME is used in the SDTM programs, the macro must be updated accordingly.

```
data LINES;
    set PROGS;
    length FILEVAR $256;
    FILEVAR=catt("&SDTMdir", ifc("&sysscp"="WIN",'\','/'), sdtmprog);
    infile dummy filevar=FILEVAR truncover end=EOF pad;
    do until(EOF); *Will read until the file is finished;
       N=sum(N,1);
       input TEXTLINE $256.;
       raw_ds = scan(scan(textline,-1," "),-1,".");
       if index(upcase(textline),"NOTE: THERE WERE") and
          index(upcase(textline),"DATA SET RAW.") then output;
       call symput('nbmatch','1');
    end;
run;
```

	SDTMPROG	N	TEXTLINE	RAW_DS
1	AE.log	354	NOTE: There were 461 observations read from the data set RAW.AE.	AE
2	CM.log	367	NOTE: There were 2585 observations read from the data set RAW.CM.	CM
3	CM.log	527	NOTE: There were 645 observations read from the data set RAW.LFM.	LFM
4	DA.log	341	NOTE: There were 1493 observations read from the data set RAW.DA.	DA
5	DA.log	387	NOTE: There were 3375 observations read from the data set RAW.SV.	SV
6	DM.log	365	NOTE: There were 425 observations read from the data set RAW.DM.	DM
7	DM.log	395	NOTE: There were 425 observations read from the data set RAW.RAND.	RAND
8	DM.log	426	NOTE: There were 425 observations read from the data set RAW.EX.	EX
9	DM.log	486	NOTE: There were 424 observations read from the data set RAW.EX2.	EX2
10	DM.log	540	NOTE: There were 425 observations read from the data set RAW.DS.	DS
11	DM.log	704	NOTE: There were 425 observations read from the data set RAW.DM.	DM
12	DM.log	742	NOTE: There were 0 observations read from the data set RAW.AE.	AE
13	DM.log	859	NOTE: There were 461 observations read from the data set RAW.AE.	AE
14	DM.log	882	NOTE: There were 461 observations read from the data set RAW.AE.	AE

Display 1. Screenshot of previous output dataset

Because a single SDTM program may reference the same raw dataset multiple times (such as DM), the macro keeps only one relevant record from each log file. It then compiles a single concatenated list of all SDTM datasets that reference each raw dataset.

At this stage, the macro merges three sources of information: the list of raw dataset names, the corresponding observation counts, and the SDTM usage list. It also generates flags to highlight specific scenarios, such as when a raw dataset is “Not used in SDTM programming”, when there was “No observations and not used in SDTM programming”, or when it is referenced in an SDTM program but is missing from the RAWdir folder.

	RAW_DS	PROGNAME_NOTE	NOBS	REASON
1	AE	AE, DM	461	
2	AEYN	Not used in SDTM programming	425	
3	BDI	QSBD	2303	
4	BDI_YN	QSBD	2306	
5	CM	CM, DM	2585	
6	CMYN	Not used in SDTM programming	425	
7	COVID	DS, SV	41	
8	COVID_IP	EX	39	
9	CSSRS_LIFE	DM, QSCS	425	

Display 2. Screenshot of previous output dataset

OUTPUT

The final output is generated as an Excel spreadsheet that summarizes the mapping between raw datasets and their usage in SDTM datasets. In addition to the dataset names and usage information, the output includes a column for user entered notes explaining reasons for not being used. Color coding is applied for ease of review: green indicates a raw dataset with zero observations that is not used in any SDTM program, while yellow indicates a dataset with at least one observation that is unused. The programmer is expected to investigate the yellow and green rows after running this macro.

For example, Figure 1 presents a scenario where the pregnancy dataset (RAW.PGFUMSTR) did not have data present in the data cut, while Figure 2 presents a case in which the dataset was available and contained at least one observation.

Raw Datasets with occurences in SDTM Programs from:

S:\StudyABC\Biostatistics\Prog\SDTM

Ran on 06AUG25

*****SDTMs need to be BATCH RUN before running this tool*****

Raw Dataset Name	SDTM Dataset(s) where the Raw was used	Number of Observations in Raw	Reason the dataset is not used
PGFUMSTR	No observations and not used in SDTM programming	0	
YBOCSMSTR	DC, DM, SV	2069	

Green: Datasets had no records and not used

Yellow: Investigate to confirm that the RAW is not needed. Put in Reason why it is not used.

Figure 1. Excel output from Search_Raw macro example

Raw Datasets with occurences in SDTM Programs from:

S:\StudyABC\Biostatistics\Prog\SDTM

Ran on 07AUG25

*****SDTMs need to be BATCH RUN before running this tool*****

Raw Dataset Name	SDTM Dataset(s) where the Raw was used	Number of Observations in Raw	Reason the dataset is not used
PGFUMSTR	Not used in SDTM programming	2	
YBOCSMSTR	DC, DM, SV	2069	

Green: Datasets had no records and not used

Yellow: Investigate to confirm that the RAW is not needed. Put in Reason why it is not used.

Figure 2. Excel output from Search_Raw macro run on the data cut which included new pregnancy data

LIMITATIONS

This macro needs to be run after all SDTM domains are mapped and the programs are run in batch with a .log file created. Additionally, the libname that reads the raw datasets should be named 'raw'. If these are not the case, it may produce inaccurate results in the output excel file. This macro also does not identify which raw variables are mapped in the SDTM dataset, it only identifies a data source.

CONCLUSION

Mapping raw clinical data to SDTM domains is a dynamic process that requires ongoing vigilance to ensure data completeness and regulatory readiness. The search_raw macro enhances the QC framework by automating the detection of unmapped data sources, particularly those that may only become relevant over time. By integrating this tool into the SDTM development workflow, teams can reduce risk, minimize manual effort, and improve the integrity and traceability of their deliverables. The RP domain case study exemplifies how automation can prevent key data from being overlooked.

REFERENCES

Murugesan, Prasanna and Thakare, Sushant. 2012. "SAS® Macro Tool to Find Source Data Sets Used in Programs." *Proceedings of the MWSUG 2012 Conference.* Available at https://www.mwsug.org/proceedings/2012/PH/MWSUG-2012-PH05.pdf

RECOMMENDED READING

- CDISC. (2022). *Study Data Tabulation Model Implementation Guide for Human Clinical Trials, version 3.4.* https://www.cdisc.org/system/files/members/standard/foundational/SDTMIG%20v3.4FINAL_2022-07-21.pdf

CONTACT INFORMATION

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

Cally Davidson
Eric Howard
Rho, Inc
Cally_davidson@rhoworld.com
Eric_howard@rhoworld.com

MACRO PROGRAM CODE

The search_raw macro code is as follows:

```
%macro search_raw(RAWdir = ,SDTMdir =, outPath =, filenam =);

   ods listing close;

   *Reading in only the SAS datasets from the RAW Dataset folder (RAWdir macro variable);
   filename PIPED pipe %if &sysscp.=WIN %then "dir /b ""&RAWdir"" "; %else "ls ""&RAWdir"" ";;
   data dtfiles;
      infile PIPED pad;
      input datasets $256.;
      raw_ds = upcase(scan(datasets,1,"."));
      call symput('nbfiles','1');
      *Selecting only sas7bdat files, no folders;
      if index(datasets,".sas7bdat");
   run;
   filename PIPED clear;


   *Obtaining all of the RAW datasets and counting the number of observations;
   libname raw "&RAWdir" access=readonly inencoding=any;
   Proc sql;
      create table DS_obs as
      select upcase(memname) as raw_ds, nobs
      from dictionary.tables
      where libname='RAW';
   run;


   *Reading in only the SDTM .log files from the SDTM program folder (SDTMdir macro variable);
   filename PIPED pipe %if &sysscp.=WIN %then "dir /b ""&SDTMdir"" "; %else "ls ""&SDTMdir"" ";;
   data PROGS;
      infile PIPED pad;
      input sdtmprog $256.;
      if index(sdtmprog,".log");
      call symput('nbfiles','1');
   run;
   filename PIPED clear;

   *If no SDTM log program files then put this warning;
   %if ^%length(&nbfiles) %then %do;
      %put WARNING: No files found in directory &SDTMdir..;
      %return;
   %end;

   *Reading in all of the .log files and creating a dataset based on each line from the LOG
        file;
   *Searches the LOG file for "NOTE: THERE WERE" lines and if it included RAW.xxxx;
   *If it included RAW.xxxx then it will output;
   data LINES;
      set PROGS;
      length FILEVAR $256;
      FILEVAR=catt("&SDTMdir", ifc("&sysscp"="WIN",'\','/'), sdtmprog);
      infile dummy filevar=FILEVAR truncover end=EOF pad;
      do until(EOF); *Will read until the file is finished;
         N=sum(N,1);
         input TEXTLINE $256.;
         raw_ds = scan(scan(textline,-1," "),-1,".");
         if index(upcase(textline),"NOTE: THERE WERE") and index(upcase(textline),"DATA SET
        RAW.") then output;
         call symput('nbmatch','1');
      end;
   run;


   *Only getting 1 record for Each combination of Raw Dataset with SDTM program;
   proc sort data=lines nodupkey out=uniqs;
```

```
   by raw_ds sdtmprog ;
run;

*Concatenate the SDTM(s) that each of the Raw datasets use into 1 record;
data DS_SDTM;
   set uniqs;
   by raw_ds;
   retain progname_note;
   if first.raw_ds then progname_note = strip(scan(sdtmprog,1,"."));
   else progname_note = catx(",",progname_note,strip(scan(sdtmprog,1,".")));
   if last.raw_ds;
   drop n;
run;

*Preparing for the merge;
proc sort data=DS_SDTM;
   by raw_ds;
run;
proc sort data=dtfiles;
   by raw_ds;
run;
proc sort data=DS_obs;
   by raw_ds;
run;

*Merging the Raw Datasets, the SDTM(s) that use each Raw Dataset,
     and the number of Obs in the Raw;
*Updating progname_note based on conditional statements to show if they were used in the SDTM
     program or is the SDTM references a Raw dataset that is not created;
data chk;
   merge dtfiles (in=raw) DS_SDTM(in=sdtm) DS_obs;
   by raw_ds;
   if raw and not sdtm then progname_note = "Not used in SDTM programming";
   if not raw and sdtm then progname_note = "No RAW dataset associated but in: "
      ||strip(progname_note);

   if nobs = 0 and progname_note = "Not used in SDTM programming" then progname_note = "No
     observations and not used in SDTM programming";
       reason = " ";
   keep raw_ds progname_note nobs reason;
run;

*If there is no RAW. files used in the SDTM .log programs it will produce this warning;
%if ^%length(&nbmatch) %then %do;
   %put WARNING: No files in &SDTMdir contain the word;
   %return;
%end;

*Creating an Excel file based on the CHK dataset to show the Raw Datasets, the SDTM(s) that
     use each Raw Dataset, and the number of Obs in the Raw;
ods excel file="&outPath.\&filenam..xlsx" style=Printer
 options(sheet_name="Raw Data Check"
            EMBEDDED_TITLES = 'on' EMBEDDED_FOOTNOTES = 'on'
            FROZEN_HEADERS ='on'   FROZEN_ROWHEADERS ='5');

 options validvarname=upcase;

 title1 "Raw Datasets with occurences in SDTM Programs from:";
 title2 "&SDTMdir.";
    title3 "Ran on &sysdate.";
    title4 "****SDTMs need to be BATCH RUN before running this tool****";

    footnote1 j=l "Green: Datasets had no records and not used";
    footnote2 j=l "Yellow: Investigate to confirm that the RAW is not needed. Put in Reason
    why it is not used.";

  proc report data=chk nowd split='#';
     column raw_ds progname_note nobs reason;
     define raw_ds        / order "Raw Dataset Name" width=20 flow;
     define progname_note / order width=115 "SDTM Dataset(s) where the Raw was used";
     define nobs          / display "Number of Observations in Raw";
```

```
         define reason / display "Reason the dataset is not used" style(column)=   {just=l};

         *Color Coding based on values;
         COMPUTE nobs;
            IF nobs = 0 and progname_note = "No observations and not used in SDTM programming"
               then do;
               call define(_row_,'style','style={background=#C5F0C5}');
            end;
            IF progname_note in ("Not used in SDTM programming","No RAW dataset associated") and
nobs > 0 then do;
               call define(_row_,'style','style={background=#FAF1B6}');
            end;
            IF index(progname_note, "No RAW dataset associated") then do;
               call define(_row_,'style','style={background=#DE6464}');
            end;
         ENDCOMP;
       run;
%mend;
```