PhUSE US Connect 2019

Paper PP02

Streamline process: To Generate SDTM Program by Automation

Hemalatha Elumalai, Symbiance Inc, New Jersey, USA

ABSTRACT

In pharmaceutical industry, submission of clinical trial data in CDISC standards is mandatory in US-FDA. Manual Standardization of clinical data is time consuming and repeated process. Automating the program creation process will reduce the time and human errors. For a small-scale organization, the standard manual process can be streamlined by preparing essential tools which can make them self-reliant and expedite the outcome with quick and less resource utilization. This poster will focus on how SDTM program generation can be automated by SAS, using standard specification and standard metadata repository, - it could be CDISC standards or Sponsor defined standards. Also, it emphasis on the details of metadata repository and how that can be efficiently used for automation. The utility generated SAS programs will be in domain wise and can be used for further customization by the user.

INTRODUCTION

CDISC SDTM is the set of specific standard requirements that should be followed in the preparation of submission datasets to regulatory authorities. As CDISC CDASH standards or standardized EDC are more widely adopted in the database design in clinical trials, SAS macros can be developed for the automated conversion to the SDTM data structure from mapping specification. This work introduces a method, that has been developed to facilitate the automation of standardized RAW – SDTM conversion. In addition to that, creating a standard metadata library for SDTM IG standards and custom domains based on therapeutic area and sponsor.

This approach we will observe more closely to these two steps:

- **Check the Mapping Specification:** Verifies the SDTM mapping specification against the SDTM IG standards using Metadata especially at variable level attributes like variable name, core, label, length & CT etc.
- **Create Mapping Programs:** Generate the sas programs based on a derivation type and assigning the attributes.

SDTM DEVELOPMENT PROCESS

Data collected from a clinical study is organized and presented using the Study Data Tabulation Model (SDTM), a data structure standard created by the Clinical Data Interchange Standards Consortium (CDISC). SDTM is beneficial for clinical data reviewers and analysts as it brings standardization in presenting the data. To ensure the quality of SDTM submission ready datasets, we must focus on the implementation of CDISC standards, such as whether the dataset contains SDTM compliant variable names, types, labels, and code lists at the time of submission.

To avoid rework in specification regarding compliance issues after generating the datasets, we have created utility based on SAS ® and MS Excel that check compliance as well as metadata issues during initial stage of SDTM development process.

SDTM SPEC CHECKS

A very decisive part of the Standardization process is defining the SDTM mapping specification; it commences with annotations of the CRF in line with the SDTM structures by analyzing the raw data. This mapping spec development is time overwhelming, doing it initial time right can save time & efforts. This will be attained by using the SDTM Spec Checks and identifies the SDTM compliance issues at spec level itself before we kick off with programming for SDTM datasets. The goal of this utility is to identify the SDTM variable level compliance issues in the first draft of SDTM specification itself to avoid errors and warning during/after programming the SDTM dataset and to avoid multiple versions of SDTM mapping specification and this can be achieved using the SDTM Spec Checking Macro.

The utility reviews the specification without editing the original specification and additionally generates compliance report (Figure 4), which will help in identifying the SDTM compliance issues in specification.

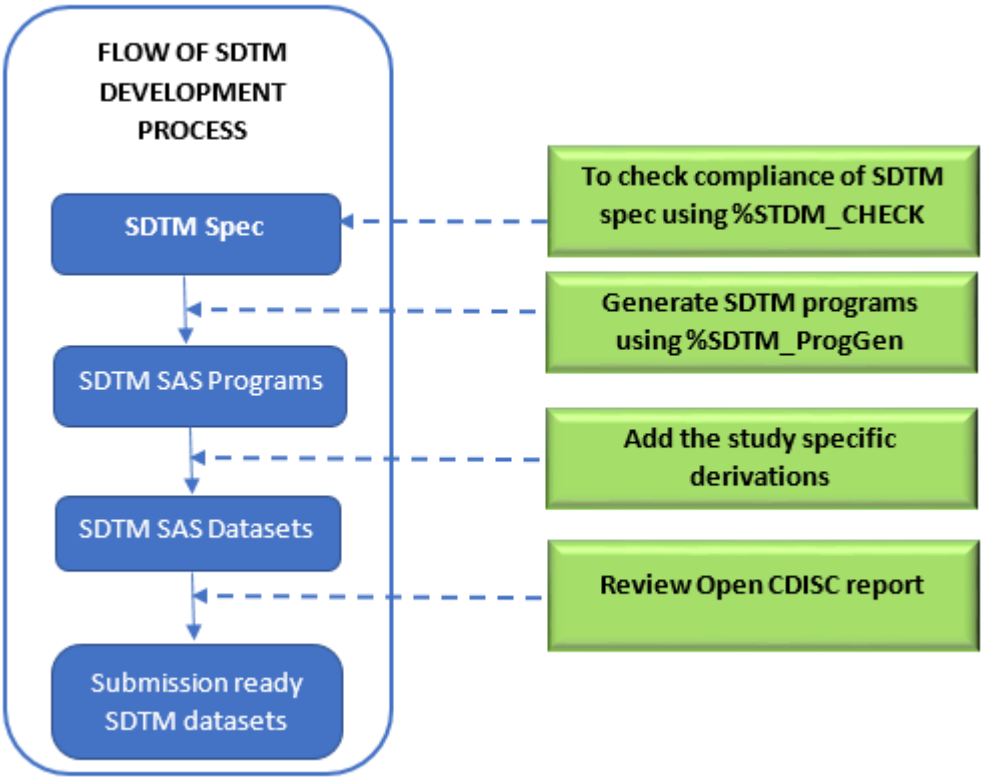

Figure.1 Traditional SDTM Process Flow

STANDARD METADATA REPOSITORY

All the SDTM IG standards and codelist will be placed in the metadata repository. So that SDTM Specification will be verified against all versions at the same time.

While a centrally managed metadata repository can accelerate the implementation of standards and facilitate regulatory compliance, it also supports reusability of standard algorithms, automation of business processes and full data traceability.

IMPORTED DATA FROM SPEC

This section will have the imported metadata from the actual specification into two sections. Domain Label, Structure, Key Variables, Sort Order

%SDTM_Check Macro:

Specification document containing all necessary information about the mapping from Source to SDTM will be used to create mapping programs automatically. Before beginning the task of developing programs to create SDTM domain data sets from your existing data, the common SDTM checks are executed first to check basic SDTM compliance and non-compliance issue. Spec Metadata compared with the standard metadata repository SDTM IG Standards with all relevant information for consistency with metadata.

In this macro all relevant information of the domain & variable level metadata are checked and compared:

- Variable name
- Type of a variable
- Length of a variable
- Codelists / Formats
- Variable Label
- Required, Expected & Permissible variables
- Not mapped variables

Macro Call for SDTM_Check

```
*****************************************************************;
/* Macro calling for SDTM Spec Check */
*****************************************************************;

%SDTM_Check (metafile= , /* Metadata Location */
             iSpecdoc= ,  /* Full path and filename of Specification document*/
             oexcploc=    /* Compare Report Location */
             );
```

The given input SDTM specification file imported using proc import, creates the two segments of specification into two datasets. The domain name, class, structure & key variables will be verified initially. Later the Variable level checks will be applied.

AE Domain Mapping Specifications								
Protocol Number:	XXXX							
SDTM Version:	3.1.3							
SDTM Domain Description:	Adverse Events							
SDTM Domain Structure:	One record per adverse event per subject.							
Program Name:	ae.sas							
General Observation Class:	Events							
Source Data Used:	AE, DM							
Key Variables:	STUDYID, USUBJID, AETERM, AESTDTC							
Sort Order:	USUBJID, AESTDTC, AEENDTC, AETERM							
Source Variable	**Target Variable**	**Variable Label**	**Datatype**	**Length**	**Controlled Terms or Format**	**Conversion Definition**	**Derivation**	**Derivation Order**
STUDYID	STUDYID	Study Identifier	Char	$200		XXXX	Assigned	
DOMAIN	DOMAIN	Domain Abbreviation	Char	$200		AE	Assigned	
USUBJID	USUBJID	Unique Subject Identifier	Char	$200		Concatenate STUDYID and SITEID and SUBJID separating by ""-"". Eg: "XXX-1001-1001"	%USUBJID	1
AESEQ	AESEQ	Sequence Number	Char	$200		Sort by variable USUBJID, AESTDTC, AEENDTC, AETERM. Restart the numbering at each USUBJID with 1 (order sequentially).	%_SEQ	3
AEACN	AEACN	Action Taken with Study Treatment	Char	$200	ACN	Direct Mapping from AEACN	Direct	
AESTDAT	AESTDTC	Start Date/Time of Adverse Event	Char	$200	ISO 8601	Concatenate the variables AESTDAT and AESTTIM. Convert and to ISO 8601 format	%DTC	2
AESTTIM	AESTDTC	Start Date/Time of Adverse Event	Char	$200	ISO 8601	Concatenate the variables AESTDAT and AESTTIM. Convert and to ISO 8601 format	%DTC	2
AEREL	AEREL	Causality	Char	$200	AEREL	Direct Mapping from AEREL	Direct	
AESPID	AESPID	Sponsor Identifier	Char	$200		Direct Mapping from AESPID	Direct	

Figure.2 AE SDTM Specification

The first step in the mapping process involves the comparison of the source metadata with the SDTM domain metadata.

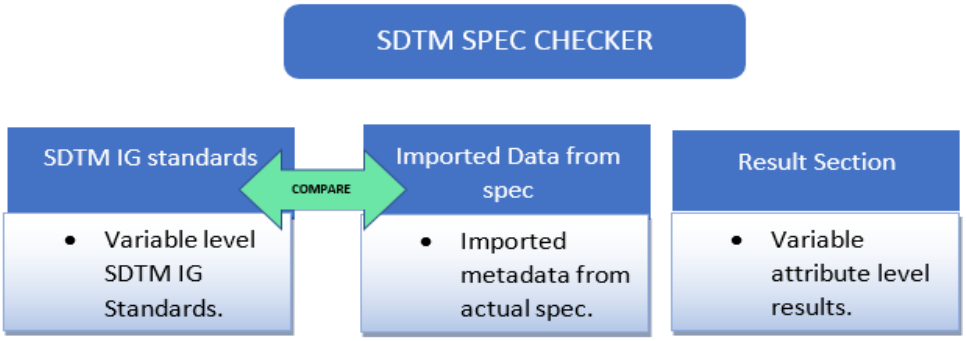

Figure.3 Basic Schematic for SDTM Spec checker

RESULT SECTION

The SDTM_CHECK macro writes an **Exception sheet** to the Excel file with information about the domains and variables. If there is any mismatch found, it will be listed in category wise. One can just filter out the domain or exception level view and verify the results against actual spec metadata and SDTM IG standards. This will help us in understanding the spec issues at one go. The resulting comparison report is the most important and it keeps the information which will be needed to update the Excel mapping specification.

An example of an **Exception** sheet within the specification file is given.

Domain	Exception Level	Exception Message
AE	Variable Level	Variable label doesn't match for these variables: AESPID
AE	Datatype	Datatype doesn't match for variables: AESEQ

Figure.4 Output from SDTM Spec checker

MAPPING SOURCE DATASETS TO SDTM DATASETS

A list of general variable mappings from raw to SDTM conversion are given below.

DIRECT - a source variable is copied directly to a domain variable without any changes other than assigning the CDISC standard label

CONVERSION - conversion of numeric to character variables

RENAME - only the variable name and label changes without any changes in the values

STANDARDIZE - standardising the result variables in Findings domains to standard units or standard terminology

REFORMAT - conversion of formats applied without any changes in data level. Like, Date Variables conversion from character to ISO8601 format

COMBINING - combining two or more variables to form a single SDTM variable

SPLITTING - source variable is divided into two or more SDTM variables

DERIVATION - creating a domain variable based on a computation, algorithm, series of logic rules or decoding using one or more source variables (e.g. __DY, __BLFL, EPOCH).

%SDTM_progGen:
The final step of this approach is the creation of the SDTM programs. These programs also called "*program Template"* because in a post-processing step the program could use these automatically generated programs as basis (template) for changes or further derivations which were made after the specification was created. Of course, changes of the derivation could also be included in the specification and the SDTM programs could be generated again. Below Code illustrate the details of how the mapping file is generated from the Excel specification by a macro **%SDTM_progGen**.

The following shows the Calling of the mapping program generation macro

```
*************************************************************;
/* Macro calling for SDTM Program Generation*/
*************************************************************;

%let gStudyID=XXX;
%let gSponsor=ABCD;
%let gUserNam = %sysfunc(propcase(&sysuserid));
%let date=%sysfunc(today(),e8601da.);


%SDTM_progGen (iSpecDoc =,/*Full path and filename of Specification document*/
               iDomains =, /* All or List of Domains with space delimited */
               oProgLoc =   /*Program output Location */
               );
```

HEADER AND GENERAL SETUP BLOCKS IN DOMAIN PROGRAMS
The program template macro **%SDTM_progGen** consists basically of a loop through all domains assigned in the macro call in the *ldomain* parameter. It is possible to call all domains given in the specification Excel file.
Loop through all domains
a. Read specification from Excel
b. Create header of template file
c. Insert derivations (Data Step / Macros) and comments (descriptions) into template file

Importing Specification and create the spec data for programming:

Variable transformations are read into a dataset with their column names to proceed the programming steps.

```
/*Count the number of Domains*/

%let i=1;
%do %while (%length(%scan(&iDomains,&i,%str( ))));
%let lDomain=%scan(&iDomains,&i,%str( ));
%let i=%eval(&i+1);

proc import datafile="&iSpecDoc"
 out=t_specdata
 replace;
      range="&lDomain$A12:K";
run;
```

Sort order Variables is imported into a separate dataset to produce the by variables order in sequence generation. The following code creates the macro variable byvar to each domain.

```
proc import datafile="&iSpecDoc"
 out=t_byvar
 replace; getnames=no;
      range="&lDomain$D10:D10";
run;
```

```
%let byvar=;
data _null_;
      set t_byvar;
      array a{*} _character_;
      call symput('byvar',tranwrd(a(1),',',''));
run;
```

The following shows the header of the mapping program generation macro

```
/*Header Block - Start*/
line="*************************************************************************"
;output;
line="Sponsor          :  &gSponsor";output;
line="Study            :  &gStudyID";output;
line="Program Name     :  &lDomain..sas";   output;
line="Purpose          :  Create SDTM+ Dataset &lDomain"; output;
line="Author           :  &gUserNam";output;
line="Date Created     :  &date";output;
line="Modification     :  ";output;
line="*************************************************************************;
";output;

/*Header Block - End*/
```

Epcoh Macro calling derivation in SDTM_ProgGen

Epcoh Calling macro is designed based on the domain class. Events and interventions are grouped together. Epoch is derived according to the __STDTC variable whereas in Findings Class, the derivation is as per the __DTC variable.

```
/*Code for MACRO derived Epoch variable in Events class*/

%let IntCount=1;
data t_macro;
      length line $1000.;
      set t_Specdata end=final;
      where upcase(Derivation_type)="MACRO";
      retain count 1;
      if upcase(derivation)="EPOCH" then do;
            line='%epoch(indset='||"&lDomain._der"||strip(put(count,8.))||
                        ", stdtc=&lDomain.stdtc,
outdset=&lDomain._der"||strip(put(count+1,8.))||");";
            count=count+1;
      end;
      if final then call symputx('IntCount',count);
run;
```

GENERATE THE PROGRAMS

SAS ® macro ***%SDTM_ProgGen*** reads the specification file and creates program files (template programs) for all domains of the clinical study in accordance with SDTM structure. The templates follow programming guidelines and adhere to program templates (e.g. program header with correct description of the program and all necessary information) used. All derivations will be incorporate in the sas program based on the given derivation type in the specification Excel file. Comments will be automatically inserted before a derivation / macro call.

Indentation is also inserted into the mapping programs by the *%SDTM_ProgGen* macro. If additional changes or study specific modifications of the derivation / mapping are necessary, the program templates could be changed.

In some cases, a specific order of the derivations needs to be followed in the program flow. This could be considered in a derivation order column where the order of the derivation steps could be changed. A derivation method describes how a variable is derived (data step/macro).

A series of small macros are incorporated to accomplish the transformation using the specification:

1. *%Usubjid* is used to derive the USUBJID variable from STUDYID, SITEID and SUBJID.
2. *%Epoch* is used to derive the epoch variable based on the __DTC variable.
3. *%dtc_cal* is a macro to transform _DAT & _TIM to __DTC variable.
4. *%_seq* is used to generate sequence number based on the sort order given in the specification.

Final steps of the programming generation ordering the variables as per the SDTM IG, assigning the attributes from the spec file which includes trimming the variable length as per the data. Keeping only the SDTM variables and assign the domain label.

Sample code for derived variables with some basic derivations done in a data step.

```
*****************************************************************;
/* Derivation of Usubjid */
*****************************************************************;
%Usubjid;

*****************************************************************;
/*Study Day Calculation */
*****************************************************************;
%dycal(in_ds=ae_der1, dtc=aestdtc, op_ds=ae_der2 );


*****************************************************************;
/*Epcoh Derivation based on Date*/
*****************************************************************;
%Epoch(in_ds=ae_der2, dtc=aestdtc,op_ds=ae_der3);

*****************************************************************;
/*Sequence number generation by the sort variables*/
*****************************************************************;
%_seq(in_ds=ae_der3, sortvar=ae_sort, op_ds=ae);
```

Sample generated program to assign attributes for the final dataset using “attribSDTM” macro

```
*****************************************************************;
/* Assigning Variable attributes */
*****************************************************************;

%attribSDTM(inpds=AE, ops=2);
```

ADAVANTAGES:

- Review the specification and identify the compliance issues at various level.
- Flexible for user to generate and update programs which reduces time and resource.
- Sub-macros can be modified specifically to deal with different projects
- Multiple macros can be integrated seamlessly.
- Spec and program consistency maintained throughout the process.
- Generated program files comply with industry and company standards and accord with good programming practice.

CONCLUSION

We have shown how to automate the process of creating clinical study data into an SDTM data. A result of this process is a mapping programs are fully generated automatically. In a further stage additional data checks could be implemented and given the user more information about required changes or also feedback about issues in the data. Also, the implementation of the define.xml documents could be easily implemented based on the complete specification. These utilities are under review at the time of submission of this paper.

REFERENCES:

1. SAS Institute Inc. 2009. SAS ® 9.2 Macro Language: Reference. Cary, NC: SAS Institute Inc.

2. Clinical Data Acquisition Standards Harmonization (CDASH) User Guide, V 1.0
   https://www.cdisc.org/standards/foundational/cdash/cdash-model-10

3. Study Data Tabulation Model Implementation Guide: Human Clinical Trials Version 3.1.3
   http://www.cdisc.org/system/files/members/standard/sdtmigv3.1.3.zip

4. Yun (Julie) Zhuo (2018), “Automate Repetitive Programming Tasks: Effective SAS® Code Generators”.
   https://www.lexjansen.com/wuss/2018/73_Final_Paper_PDF.pdf

CONTACT INFORMATION:

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

Hemalatha Elumalai
Symbiance Inc. 51 Everette Dr,
Suite B-20, Princeton Junction
New Jersey 08550, USA
Email: helumalai@symbiance.com
Web: www.symbiance.com

Brand and product names are trademarks of their respective companies.