PharmaSUG 2024 - Paper SD-198

AutoSDTM Design and Implementation with SAS Macros

Chengxin Li, AutoCheng Clinical Data Services LLC

ABSTRACT

CDISC standards have been widely deployed in the pharmaceutical industry for the past decade. The main effort has focused around complying with CDISC standards. While complying with CDISC standards, the focus should shift toward developing a reusable automated solution.

The metadata of raw data varies across companies. However, the metadata patterns are limited and the metadata of targeted SDTM are rigid defined in SDTMIG. This provides the feasibility of SDTM mapping automated. This paper introduces a universal autoSDTM solution implemented with SAS macros.

Design goals 1. No dependence on data management settings thus a universal solution; 2. Besides current SDTMIG 3.3 compliance, output with submission ready; 3. Process transparent to facilitate review; 4. Following SDTM programmer's practices thus with low transition cost; 5. User friendly; 6. Most importantly, high efficiency while ensuring high quality.

Implementations 1. One SAS macro per domain; 2. Macro parameters aligned with the SDTM variables, thus parameterizing and getting; 3. Input conventions and checking for dataset name, variable name, format and informat, and outcomes to SAS logs of note, warning and error for user friendly; 4. Outputs including SDTM dataset, LOG, mapping specifications, and plain SAS executable codes dynamically generated per domain; 5. Control terminology portal enabled; 6. For special cases, data preprocessing is supported.

It's verified that the above autoSDTM solution is practical and of high quality and efficiency. With the solution, not including trial design domains, a total of 50 SDTM domain macros have been developed. The tool has been validated in multiple studies and applied to client productions.

INTRODUCTION

Clinical Data Interchange Standards Consortium (CDISC) standards have been widely deployed in the pharmaceutical industry for the past decade. Mainly, much of the effort has focused on complying with CDISC standards. While compliance with CDISC standards is important, the focus should move beyond compliance and shift toward developing a reusable automated solution.

From a programming perspective, the clinical data processing automation could include, but is not limited to, CRF annotation (for SDTM), SDTM mapping, ADaM development, and TLF generation. Currently, SAS macros are still the primary tools being used in the pharmaceutical industry. The focus of this paper will be on SDTM development automation implemented with SAS macros.

SDTM domain development is resource-intensive and challenging routine work requiring more experienced statistical programming. Therefore, SDTM development automation is greatly beneficial in saving time and resources while ensuring quality. SDTM development requires fully reflecting the collected data with limited derivations (but no imputations), or transformations from horizontal to vertical structure. The collected raw data is usually from multiple sources, CRF collected or external transferred. Meanwhile, the metadata setup for the collected raw data is very much varied, even implemented with CDASH-like standards. Those factors bring up the challenges on developing one universal automaton tool applicable to the productions.

Although the raw data metadata are dynamic, the metadata patterns are very limited. Most importantly, the targeted SDTM domains have very rigid metadata defined in the SDTMIG, implementation variations still existed in productions though. From the varied to the rigid can be bridged by parameter settings with SAS macro, i.e., the different metadata patterns and common implementation variations can be incorporated into different options of SAS macro parameters for a user to choose. The macro parameters can be further harmonized with standard variables defined in SDTMIG per domain. This provides the feasibility of SDTM automated and implemented with SAS macros. Correspondingly, the programmer can switch the operations from routine programming to macro parameter settings. All other work is performed by the macro aligning with the parameter settings. The parameter setting functions as mapping specification, thus parameterizing and getting. The approach is named autoSDTM driven by SDTMIG.

This paper introduces the design and implementation for SDTM development automation implemented with SAS macro. Macro %DM for the special purpose domain DM generation is selected to illustrate the autoSDTM design and implementations. The current version conforms to SDTMIG 3.3 in terms of dataset structure and variable metadata.

GOALS AND METHODOLOGY

As specified, SAS macro is utilized to develop the SDTM automation tool aiming to improve quality, efficiency, and cost reduction in SDTM developments. Some general considerations are depicted in this section.

The developed autoSDTM macros function as a mapping tool to fully reflect the collected data without any imputations although limited derivations/conversions and controlled terminology alignments are acceptable. In production, the macro shall not function as a data correction tool. The user shall report any data issues though the query process.

1. OBJECTIVE

Besides high efficiency while ensuring high quality, it's targeted to develop a universal automated SDTM development tool and the SAS programmer as the primary user. Table 1 summarizes the design goals.

Table 1 Design Goals

Objective	Requirement
Process automated	While keeping the specification working scope, the routine programming work handled by the tool
Universal solution	No dependence on raw data metadata settings, thus working with any clinical metadata formats from unfavorable legacy to preferable CDASH; correspondingly, also applicable to small and medium-sized companies; client needs-oriented
Programmer as the primary user	Following current industry programming style thus with low transition cost
User experience friendly	process transparent, eliminating black box
Best practice incorporated	Complying with GCP and programming SOP, e.g., programming hard code prohibited; Not overwriting/skipping functional processes such as the unblinding process, treatment group switching process due to med error, etc.
Output with submission ready	Domain output in SAS and XPT format; metadata compliance with SDTMIG; Control Terminology aligned; DEFINE-XML generation facilitated

User experience is an important general consideration in developing a tool. Below section further details this topic.

2. USER EXPERIENCE

The developed tool should be easy to understand and easy to use. With that in mind, Table 2 summarizes the factors along with the corresponding method that attempts to improve the user experience.

Table 2 Factors Considered in Enhancing the User Experience

Factor	Method
Platform	Multiple platforms supported, avoiding platform-specific coding, or dynamically coding as appropriate
Parameter Name	1. Straightforward and intuitive to understand 2. Aligned with SDTMIG standard variables
Parameter input and format	1. Consistent input conventions on parameters across macros such as using the "#" to denoted multiple items, "\" for separated subcomponent 2. Consistent input format to support patterns of assigned value, directly mapping from raw data like DSNM.VARNM, and term/order converts with predefined SAS FORMAT/INFORMAT 3. VARTYPE in char or numeric handled by the macro without user denoting 4. User inputs checked automatically and the outcomes to SAS LOG, for instance,

	existence checking for dataset, variable, and SAS format/informat, thus easily identifying input issues such as typos
LOG	1. User operation outcomes with SAS NOTE, WARNING, ERROR, and 2. Saved as an external file and transparently replayed back to the SAS LOG Window 3. LOG becoming the communication channel between the user inputs and outcomes
Specification	1. Mapping specification automatically generated aligning with macro parameter settings 2. Facilitating review and cross-checking either with available mapping specs or with which further producing the mapping specs 3. Facilitating CRF annotations
SAS Codes	From the macro call instance, the corresponding plain SAS executable codes are automatically generated, with which the same SDTM can be produced
User Manual	1. Instructions pre-coded inside using SAS NOTE, WARNING and ERROR 2. Manual (readme) in PDF further produced

With the above methods, keeping transparent with macro parameter configurations, a user can achieve **seeing and knowing**, **parameterizing and getting** in SDTM development.

3. DEVELOPMENT LIFECYCLE AND STRUCTURED PROGRAMMING

Another general consideration is to use the basic software engineering principles following a standard development lifecycle and a structured sequential design. The development lifecycle defines several concrete steps after which feedback is provided to the previous step. Structured design is a methodology that divides a complex task into smaller modules. These modules are logically arranged into linear sequential coding blocks.

The autoSDTM development lifecycle is depicted in Figure 1 consisting of concept design, implementation, testing, documentation, and production. During the concept design stage, details are documented as clearly and thoroughly as possible to define the scope, methods, and parameters. After each step, feedback is provided to the preceding step. Before the macro enters production, the macro should be fully tested, and supporting documents such as a parameter dictionary or user guide should be available.

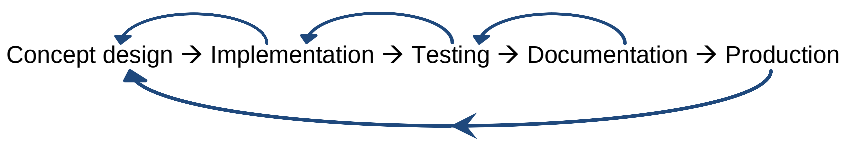

Figure 1. autoSDTM Development Lifecycle

The autoSDTM development leverages the structured design and sequential modules. This allows for organizing codes into sections that achieve specific objectives. Modular coding is easier to maintain since any issues that may appear can be isolated to a particular module. It is also easier to reuse since the modules can be swapped in and out depending on the macro being developed. Table 3 summarizes the five sequential programming modules used in each of the individual macros.

Table 3 Sequential Programming Modules of autoSDTM Development

Module	Objective	Description
1	**Initiation**	1) Set system options 2) Record the initiation status (used later during cleaning-up) 3) Define sub-macros 4) Build up dataset templates
2	**Checking**	1) Check the existences of LIBREFs, datasets, variables, formats and informats 2) Check user input conformance with defined rules and requirements 3) Prepare for derivations, e.g., along with checking the validity of macro parameters, parse the parameters so that they are derivation-ready

Module	Objective	Description
		4) Write the outcomes of checks to SAS LOG window 5) Generate mapping specs
3	**Derivation**	1) Perform variable derivations by user parameter settings 2) Resolve the macro to plain SAS executable codes 3) Prepare the dataset and associated documents for output
4	**Output**	Store SDTM dataset and associated documents in the designated folder
5	**Cleaning-up**	Reset settings to the status before the macro call, deleting or keeping intermediate datasets

DESIGN AND IMPLEMENTATION

The structured design and sequential modules have been leveraged in developing autoSDTM tools with SAS macros. Since SDTM development transforms the varied raw data to the rigid SDTM domain, the macro parameters are aligned with SDTM standard variables, i.e., one macro parameter linked to one or a group of SDTM standard variables; a separate macro parameter linked to all the non-standard variables. Meanwhile, different raw data patterns are incorporated into the macro parameters. Therefore, the configuring macro parameters are harmonized with mapping specification process. Equivalently, macro parameters and the corresponding settings are leveraged to the mapping specifications. The method is also named autoSDTM driven by SDTMIG.

For easy operations, it’s designed as one parameter associated with one domain variable, one SAS macro aligned with one SDTM domain. For the same purpose on easy operations, under some scenarios, one SDTM variable may also be associated with several macro parameters, e.g., SE.ETCD with PRIOR, TRT, and POST by EPOCH; DSTERM with PROTMS, DISPEVT, and OTHEVT by DSCAT.

In SDTMIG, SDTM production models are classified as special purpose domains, general observation classes (interventions, events, and findings) and relationships. While each SDTM class or dataset has its own specific considerations, there are some common points across these domains, most of which are implemented with utility macros. Table 4 lists the common factors specifically addressed.

Table 4 Common Factors in Developing autoSDTM Tools

Factor	Method	Note
ISO8601	1. Four date patterns auto-identified and converted: MM-DD-YYYY, DD-MM-YYYY, YYYY-MM-DD, and DD-MON-YYYY; special case like 5-2-2020 also incorporated and supported 2. Full or partial missing handled	All the *DTC variable
Var Split	1. If CMDECOD from WHO-DD >$200, then split by a semicolon with each closest to $200, storing the first to CM.CMDECOD, others to SUPPCM. 2. If COVAL > $200, then split by words with each closest to $200, storing the first to CO.COVAL, others to CO.COVAL1 to COVALn	CM and CO
Var Length	Char variables length optimized within the domain; optionally, some common char variables length optimized across domains, e.g., EPOCH length aligned with the max length of TA.EPOCH	All the domains
VISIT/VISITNUM	1. SV holding all the VISIT/VISITNUM per subject. Unscheduled Visit derived by applying parent-child relationship To decrease the chance of MxM merge, for a domain: 2. Scheduled Visit derived from TV 3. Unscheduled Visit derived from SV	Applying to most finding domains

EPOCH	Along with TAETORD, EPOCH is derived from SE, and SE.SEENDTC of last ELEMENT filled with DM.RFPENDTC	With SDTMIG3.3 defined; TAETORD not included in DS
--LOBXFL	Last non-missing value prior to RFXSTDTC; if both --DTC and DM.RFXSTDTC containing time part, then compare with full DTC, otherwise only with Date part. Along with by --TESTCD, other variables also possibly added to By Vars, e.g., LBCAT, VSPOS	--BLFL replaced
SUBJID	Besides DM, optionally added to other domains; correspondingly, multi-SCRN or multi-Enrolled subjects with clear path per domain	Requested by Study Data Technical Conformance Guide - 4.1.1.2[1]

The autoSDTM tool takes --BLFL as a retired variable and replaced with --LOBXFL in clear definitions; --DRVFL not performed by the tools, i.e., in mapping, additional records will not be derived by programming such as AVERAGE/AGGREGATE if not in raw data. If analysis is requested, ADaM would be the right programming spot to do so with SAP.

Very often, in finding domains, the source database is structured in a denormalized(horizontal) structure, and the source variable named in variations. To cover all the scenarios, either normalized or denormalized, the macros provide an option by creating an intermediate variable to link the result variable names to the TESTCD.

The simple derivations can be performed by the macro (similar to one or multiple IF…THEN statements). However, for a complex derivation, sometimes data processing might be needed before the macro call. Additionally, it's beyond the macro's capability to dynamically subtract one requested variable from another source variable for mapping. For instance, LOC, LAT, and DIR are often combined into one collected raw variable. It's preferable to separate those components into several targeted variables --LOC, --LAT, --DIR before the macro call rather than handled by the macros. The autoSDTM supports the data preprocessing option by loading the topic dataset from WORK libref instead.

There are input conventions defined for macro parameters. Those macro parameter input conventions and rules have been pre-coded insider of macros. The macro performs the input checking. If not conformed, the messages will be printed in LOG window and saved in a log file. Table 5 lists the common separators reserved in parameter inputs.

Table 5 Symbols Reserved in Input Parameters

Symbol	Description
#	separating multi-items in a parameter, e.g., AGE=AGE # AGEU='YEARS'. For a parameter, unless specified, multiple components supported and separated with #
.	denoting DSNM and VARNM, e.g., EX.EXSTDAT
|	concatenating two components, e.g., EXSTDAT|EXSTTIM
\	separating two parallel subcomponents, e.g., EX.EXSTDAT\WHERE
() or [] or {}	grouping multiple source variables for one target variable, e.g., RACE=(AMERI ASIAN BLACK NATIV WHITE RACEOTH)
KEYWORD	Keywords used in parameter inputs: FORMAT, INFORMAT, FORMULA, QNAM, QLABEL

With autoSDTM tools, the mapping specification is moved up to the macro parameters, correspondingly parameterizing and getting. Table 6 describes the outputs from the corresponding macro call with the user parameter settings.

[1] https://www.fda.gov/media/153632/download

Table 6 autoSDTM Output

Output	Description
SDTM Domain--XX	XPT and SAS datasets, metadata (names, labels, orders) in compliance with SDTMIG, variable length optimized
SUPP/CO Domain--SUPPXX/CO	Same as above; additionally, QVAL in date converted to ISO8601 format, and N/Y also aligned
SAS Codes--xx.txt	Plain SAS executable codes resolved from the macro call instance; execute xx.txt producing the same dataset as the macro call
SAS LOG--XX.log	SAS log file, and replay back to SAS LOG window transparently
DATASETS/CODELISTS/ VARIABLES	Three SAS datasets, aggregated per domains for mapping specification, and templated with the common compliance checking tool. In VARIABLES dataset, besides mapping specification, annotations are also specified to facilitate CRF annotations. With a utility macro, the datasets can be converted to an excel file with either horizontal multiple sheets for one domain per sheet or vertical one sheet holding all domains.

Figure 1 summarizes the autoSDTM designs and implementations. The user parameterization is to link one or multiple source variables to the SDTM variable by incorporating the source variable metadata patterns, the same process as the mapping specifications. If a new pattern appears, just add it like training. However, for a rare random case, it may comprise between the complexity and data pre-processing.

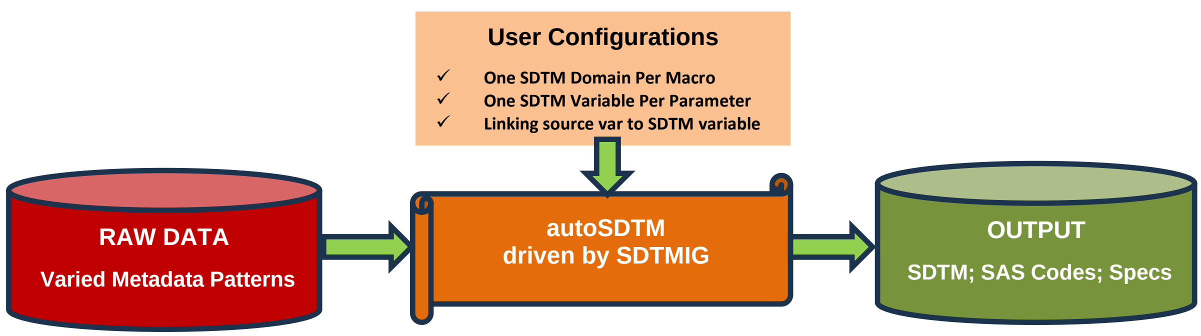

Figure 1 autoSDTM Designs and Implementations

In the EXAMPLE section, DM macro (%DM) is depicted to illustrate the autoSDTM concepts.

%DM EXAMPLE

Operation dependency exists in domain development. Trial design domains, TA/TV/TI, are needed for corresponding subject domain generations. Among those subject domains, special purpose DM domain generation is always the first, then SE and SV, and following any other general observation classes. DM, SE and SV provide dependencies to develop other domains: DM.RFSTDTC for *DY derivations; SE.SESTDTC/SEENDTC for EPOCH derivations; and SV.SVSTDTC/SVENDTC for VISIT and VISITNUM derivations, esp. for unscheduled visits.

Table 7 lists %DM macro parameters and definitions, named as parameter dictionary table with columns of parameter name, input designation, definition, and note/example.

Table 7 %DM Parameter Dictionary

Parameter	Core	Definition (# separated for multiple items)	Note/Example
RAWLIB	req	Single raw data libref name	%str(RAW) Note: applied to all macros
SDTMLIB	req	Single SDTM output libref name	%str(SDTM) Note: applied to all macros
MISCLIB	req	Single MISC libref name storing study misc datasets to access	%str(OTHER) Note: applied to all macros (opt)
TOPICDS	req	DSNM, raw dataset containing TOPIC variable	%str(DEMO) Note: applied to all macros
SUBJID	req	topic variable name, format: DSNM.SRCVAR # DSNM.PRESBJVAR\WHERE DSNM.PRESBJVAR\WHERE supporting the case of multi-SCRN/multi-ENRL and the previous subject ID mapped to SUPPDM by default	%str(SUBJECT) Note: it requires the source var of subject ID existed in all the raw datasets per study, e.g., SUBJECT; or, derive one
DMVARS	req	format1: SDTMVAR=DSNM.SRCVAR, list of direct mapping variables incl domain specific IDs and domain specific vars, DSNM optional with topic dataset as default, for DM: STUDYID, SITEID SEX, INVID, INVNAM, RACE, COUNTRY, ETHNIC, AGE, AGEU format2: SDTMVAR="XXX", for value defined in protocol but not in CRF/Rawdata format3: SDTMVAR=SRCVAR1 SRCVAR2 … SRCVARn, applied to RACE, denormalized/horizontal structure in raw data, by default, label transposed to value; if multiple components selected, then with MULTIPLE in RACE, components stored in SUPPDM	%str(STUDYID=STUDY # SITEID=SITEID # SEX=SEX # RACE=RACE # COUNTRY="USA" # ETHNIC=ETH # AGE=AGE # AGEU="YEARS")
USUBJID	req	format: separated with | as input, two or three ID variables concatenated, or single DSNM.USUBJIDVAR (as collected) for a rollover/extension study	%str(STUDYID|SITEID|PTNO) %str(STUDYID|SUBJECT) %str(USUBJID)
RFSTDTC	req	format: DSNM.DATVAR|DSNM.TIMVAR\WHERE, TIMVAR and WHERE optional, providing source vars to derive RFSTDTC (min)	%str(EX.EXSTDAT|EX.EXSTTIM) Note: RFSTDTC is study specific
RFENDTC	req	format: DSNM.DATVAR|DSNM.TIMVAR\WHERE, TIMVAR and WHERE optional, providing source vars to derive RFENDTC (max)	%str(EX.EXENDAT|EX.EXENTIM) Note: RFSTDTC is study specific
RFXSTDTC	req	format: DSNM.DATVAR|DSNM.TIMVAR\WHERE, TIMVAR and WHERE optional, providing source vars to derive RFXSTDTC (min)	%str(EX.EXSTDAT|EX.EXSTTIM)
RFXENDTC	req	format: DSNM.DATVAR|DSNM.TIMVAR\WHERE, TIMVAR and WHERE optional, providing source vars to derive RFXENDTC (max)	%str(EX.EXENDAT|EX.EXENTIM)
RFPENDTC	req	Format1: DSNM.DATVAR|DSNM.TIMVAR, TIMVAR optional, providing source vars (referring to protocol flow chart) to derive RFPENDTC (max); Format2: NULL or RFPENDTC\ADDVAR=\EXLVAR=, requesting %RFPENDTC to scan all date variables (DATE, DAT, DT), optionally with additional variable	%str(RFPENDTC\ADDVAR='LBDTM')

		added and/or variables excluded; to facilitate review, the temporary RFPENDTC dataset produced, listing the max DTC value and the associated source DSNM and VARNM per subject	
DTHDTC	opt	format: DSNM.DATVAR|DSNM.TIMVAR\WHERE, TIMVAR\WHERE optional, providing source vars to derive DTHDTC and DTHFL; If the parameter not provided, DTHDTC and DTHFL populated with null	%str(DD.DSSTDAT # AE.AESTDAT\AETOXGR_STD=**5** or AEOUT='Fatal' # AE.AESDTH\AESDTH='Y' # EOT.DSDAT\DSTERM='Death' # EOS.DSDAT\DSTERM='Death' # SFU.SSDAT\SSORRES='Dead')
BRTHDTC	opt	format: YY=BRTHYY or YY=BRTHYY|MM=BRTHMO|DD=BRTHDD|TM=BRTHTIM, or just DT=BRTHDAT even only year collected; if not provided, then not populated	%str(DT=DM.DOB)
RFICDTC	req	format: DSNM.DATVAR (min)	%str(IC.ICDAT) Note: if multi-SCRNs, usually multi consent records in DS
DMDTC	opt	format: DSNM.DATVAR\WHERE if collected or derivable, DSNM\WHERE optional; if not provided, then not populated	%str(DMDAT)
ARMCD	req	format: DSNM.VARNM\WHERE, providing source var to assign ARMCD, DSNM stored in MISCLIB and usually derived from RAND file	%str(TRTARM.ARM_Code)
ARM	req	format: DSNM.VARNM\WHERE, providing source var to assign ARM	%str(TRTARM.ARM)
ACTARMCX	opt	format: LIBNAME.DSNM, allocate Unplanned Treatment in ACTARMCD, if any. DATASET(DSNM) requiring USUBJID and/or SUBJID and/or source VARNM of SUBJID, ACTARMCD and ACTARMUD.	%str(MISC.TRTCX) Note: only supporting dataset option to avoid subject ID hardcoding
SUPPDM	opt	format: DSNM.VARNM\QNAM='...'\QLABEL='...', list of all non-standard variables mapped to SUPPDM, QNAM/QLABEL optional. QLABEL for reassigning QNAM label, by default, the label of QNAM used; similarly to QNAM. DSNM also optional if the VAR from topic dataset	DM.PGNAIVE_STD\QNAM= "PGNAIVE"\QLABEL="Prostaglandin Analogs Naive") Note: SUPPXX applied to all macros
NSUBMIT	opt	format: DSNM.SRCVAR, facilitate CRF annotations listing [NOT SUBMITTED] variables if any	Note: may not applied to DM; applied to observation class domain macros
STDCX	opt	format: SDTMVAR\format=xxx, portal used for CT alignment	%str(SEX\format=$SEX. # RACE\format=$RACE.) Note: applied to all macros
DEBUG	opt	debugging mode to keep intermediate datasets in WORK lib if Y, N by default	%str(Y) Note: applied to all macros

For ARM/ARMCD mapping, one dataset is usually programming derived from the randomization file and stored to parameter MISCLIB folder. ARMNRS is derived by default. ACTARMUD becomes a required variable in the provided dataset with ACTARMCX parameter.

For a subject with multiple screenings/enrollments, the primary subject ID is stored in DM.SUBJID. There are three common approaches to store additional subject ID(s): SUPPDM, XM (Custom Demographic Domain), or DC (Demographic as Collected). Considering compliance and simplicity, the autoSDTM tool uses SUPPDM option. If a SUBJID IN raw dataset but NOT IN DM domain, the autoSDTM will auto-scan the SUBJID from SUPPDM.QVAL WHERE QNAM=PRESUBID to get the associated USUBJID. The XM/DC domain is developed on request.

Table 8 lists the outputs from %DM with descriptions.

Table 8 Outputs from %DM Macro

OUTPUT	NOTE
DM	dm.xpt, dm.sas7bdat
SUPPDM	dm.xpt, dm.sas7bdat, if any SUPPQ mapped, additionally including the components of RACE=MULTIPLE and previous subject ID if a subject with multiple screenings/enrollments
SAS Codes	dm.txt. plain SAS executable codes dynamically resolved from %DM macro call instance; same DM/SUPPDM dataset produced by executing dm.txt
LOG	DM.LOG, besides the SAS LOG displayed in SAS LOG window, also saved as a physical file
DEFINE Template File	DM/SUPPDM metadata accumulated to SAS datasets of DATASETS, VARIABLES, CODELISTS With available utility macro, specs converted to excel sheet, either in one sheet or multiple sheets by DOMAINs
Mapping and Annotation SPECS	SAS datasets – VARIABLES, SDTM mapping specifications produced and also annotation text generated to further facilitate autoACRF

Following the available existing mapping specification, Figure 2 below illustrates one %DM call instance showing what the macro call looks like with programming notes added.

```
%DM(RAWLIB=%str(RAWDATA),
    SDTMLIB=%str(SDTM),
    MISCLIB=%str(MISC),

    TOPICDS=%str(DEMO),
    SUBJID=%str(Subject),
    USUBJID=%str(STUDYID|SUBJECT), /*STUDYID|SITEID|PTNO, USUBJID, Project|Subject, ...*/

    DMVARS=%str(STUDYID=DEMO.STUDYID # SITEID=SiteID # AGE=AGE #AGEU='YEARS' # SEX=SEX_STD # RACE=(AMERI ASIAN BLACK NATIV WHITE RACEOTH) #
                ETHNIC=ETHNIC # COUNTRY="USA"),

    RFSTDTC=%str(EX.EXSTDAT),  /*RFSTDTC with project specific defination*/
    RFENDTC=%str(EOT.DSSTDAT), /*RFENDTC with project specific defination*/
    RFXSTDTC=%str(EX.EXSTDAT|EX.EXSTTIM),
    RFXENDTC=%str(EX.EXENDAT),
    RFPENDTC=, /*here NULL, thus requesting macro to autoscan all datasets in RAWDATA for maximum date*/

    DTHDTC=%str(DD.DSSTDAT # AE.AESTDAT\AETOXGR_STD=5 or AEOUT='Fatal' # AE.AESDTH\AESDTH='Y' # EOT.DSSTDAT\DSTERM='Death' #
                EOS.DSSTDAT\DSTERM='Death' # SFU.SSDAT\SSORRES='Dead'), /*here lists all sources (esp. for ongoing study even having DTH CRF page)*/
    BRTHDTC=%str(DT=BRTHDAT),   /* YY=BRTHDAT or YY=BRTHYY|MM=BRTHMO|DD=BRTHDD|TM=BRTHTIM or DT=BRTHDAT|TM=BRTHTIM*/

    RFICDTC=%str(IC.CONSNDAT),  /*req., format: DSNM.DATVAR, taking minimum*/
    DMDTC=%str(VIS.VISDAT_RAW\FolderName="Visit 1"),

    SUPPDM=%str(RAND.RAND_ID\QNAM="RANDNO"\QLABEL="Randomization ID" # RAND.RANDOMIZED_AT\QNAM="RNDTGMT"\QLABEL="Randomization Date/time"),
           /*macro autoconvert date or date/time type QVAL to ISO8601 DTC in SUPPDM*/
    ARMCD =%str(TRTARM.ARMCD), /*TRTARM is a programming derived dataset located in MISC*/
    ARM =%str(TRTARM.ARM),

    ACTARMCX=%str(MISC.TRTCX), /*due to wrong med, a subject switched treatment group, TRTCX converted from study doc*/
    STDCX=%str(SEX/format=$SEX. # RACE\format=$RACE.), /*SEX and RACE CTed with SAS formats $SEX and $RACE*/

    DEBUG=%str(Y)
    );
```

Figure 2 %DM Macro Call Instance

With macro parameter settings, equivalent to mapping specifications, after executing the macro call, the user can get SDTM domain done. The settings can be reused with possible adjustments to other studies. Standardizations with metadata of raw data can make the parameter configurations more reusable.

CONCLUSION

It's verified that the above autoSDTM solution is practical and of high quality and efficiency. With the solution, not including trial design domains, a total of 50 SDTM domain macros have been developed. The tool has been validated in multiple studies and applied to client productions.

For user friendliness, a graphic interface can be further developed to simply the parameter inputs.

CONTACT INFORMATION

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

Chengxin Li
AutoCheng Clinical Data Services LLC
Chengxin.Li@autoclindata.com

Any brand and product names are trademarks of their respective companies.