

Optimizing SDTM Specification Development with Auto-Population

Cori Kramer and Ryan Adalbert, Covance Inc., King of Prussia, PA

Introduction

On many studies, the development of SDTM specifications can be optimized using SAS macros. During the creation of the SDTM data, the author manually identifies mapping rules and a certain level of metadata, for which CDISC review software or sponsor created specs can be used. When the raw data has several levels of SDTM compliance, SAS macros can be utilized to compare raw data values to the CDISC Controlled Terminology database. Macros can also check the conformance of raw values and metadata with SDTM standards. This enables the auto-population of mapping rules into the specification, significantly reducing the manual work time. Furthermore, these benefits can be achieved by using a single SAS macro that leverages MS Excel such that the spec authors never need to open raw data sets or terminology files.

Major Areas of Efficiency

Eliminates the majority of manual cross checks with CDISC Controlled Terminology

Significantly reduces the manual input of text into the specification

Increases consistency and accuracy of mapping notes



Leaves only a portion of the manual work to be done

Specification Completion Process

SAS macros can be utilized to compile details of the variables from each raw dataset in a user friendly excel spreadsheet. This allows the spec author to simultaneously view and filter through the raw variables, values, and certain metadata from dozens of datasets. The spec author can use excel to filter the information by domain and can gain an overview of the raw dataset structure and where changes may need to occur. This initial overview also will ensure that no raw variables are overlooked or missed during the SDTM development process.

In the same spreadsheet, the spec author can then view unique raw values next to the appropriate Controlled Terminology. Sponsor CT can be added where necessary and values that do not directly match up to an existing CT value can be matched. This allows the spec author to see which values are compliant. The spec author can also enter custom mapping instructions for each value that they wish to have populated in the spec.

Next, the spec author indicates which variables they want to map and how the mapping should be performed. Single character identifiers will be used to specify which pre-programmed notes should output in the specification. Some examples could be identifying where dates need to be changed to ISO8601 format, identifying where values need to be updated to match CDISC CT, or simply straight mapping variables from raw to SDTM where they already conform to the proper SDTM values and format.

Macros then read in the single character identifiers alongside the Controlled Terminology comparison to output the mapping notes in the specification. A high level check of the raw data can be done to ensure no structure issues have been missed and to populate any manual notes that are needed.

Understand Raw Data

Check Controlled Terminology

Check Raw Values for Conformance

Populate Specifications

When can we use this process?

- ▶ When up versioning SDTM, this is a very efficient way of ensuring the previous data is conformant to current SDTM and CDISC Controlled Terminology.
- ▶ When raw data has all or some fields the same as SDTM variable names, this process can save time ensuring that the raw variable data fits correctly into the SDTM variable with the same name.
- ▶ Macros can also be used to save effort identifying where each raw variable will end up in SDTM. Having the unique values all in one place helps identify source to target information.

Domain	Variable	Search	Domain	Variable
VS	DOMA	☒ (Select All)	DM	BRTHDTC
VS	STUDY	☒ DM	DM	DMDTC
VS	USUBJ	☒ DS	DM	DTHTDTC
VS	VISIT	☒ EX	DM	RFENDTC
VS	VISITN	☒ SUPPDM	DM	RFSTDTC
VS	VSBLEFI	☒ SUPPOS	DS	DSSTDTC
VS	VSCAT	☒ SUPPEX	DS	DSSTDTC
VS	VSDRV	☒ VS	EX	EXENDTC
		OK Cancel	EX	EXSTDTC
			VS	VSDTC
			VS	VSRFTDTC

Figure 1. A quick glance at the Excel spreadsheet output by the macro shows dataset names and fields very similar to SDTM. This type of data is where the optimization power of the macro is maximized. The filters can be used to look at individual domains as a whole or similar variables across domains.

Domain	Variable	Uniques	Code	Codelist_Code	Codelist_Name	CDISC_Submission_Value
VS	VSPOS	NOT DONE (AFTER 5 MIN REST)				null
VS	VSPOS	SITTING (AFTER 5 MIN REST)				SITTING
VS	VSPOS	STANDING (AFTER 5 MIN REST)				STANDING
VS	VSPOS	SUPINE (AFTER 5 MIN REST)				SUPINE
VS	VSPOS	SITTING	C62122	C71148	Position	SITTING
VS	VSPOS	STANDING	C62166	C71148	Position	STANDING
VS	VSPOS	SUPINE	C62167	C71148	Position	SUPINE

Figure 2. The Excel sheet lists the unique variable values that match identically with CDISC controlled terminology. The author can use excel to sort and filter by any of the domains or variables. Controlled Terminology values can also be entered by the author to indicate that the macro should write a case statement standardizing these values.

Domain	Variable	Uniques	Domain	Variable	Straight_Map
VS	VSDTC	yyyy-mm-dd	VS	VSDTC	Y

Figure 3. The excel sheet shows the unique format of date fields as an example of conformance. When dates are already in ISO 8601 format, the author can enter a Y to indicate that the date can be mapped as is. In this case, the raw variable will be entered into the programming notes of the spec.

Dataset	Variable	Programming_Notes
		When VS.vspos = 'NOT DONE (AFTER 5 MIN REST)' set to missing
		When VS.vspos = 'SITTING (AFTER 5 MIN REST)' set to 'SITTING'
		When VS.vspos = 'STANDING (AFTER 5 MIN REST)' set to 'STANDING'
		When VS.vspos = 'SUPINE (AFTER 5 MIN REST)' set to 'SUPINE'
VS	VSPOS	Otherwise VS.vspos

Figure 4. Programming and mapping notes are output into the spec when the macro is run. In this example, the case statement from Figure 2 is output. The manually entered matches are in the when conditions of the case statement and the remaining values are mapped as is.