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. In 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

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 variable. The 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 for 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 end up in SDTM. Having the unique values all in one place helps identify source to target information.

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.

Understand Raw Data
Check Controlled Terminology
Check Raw Values for Conformance
Populate Specifications

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.

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.

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.

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.