Paper SAS4416-2020

Data Standardization using SAS® Health: Data Mapper

Eric Bolender, SAS Institute Inc., Cary, NC

ABSTRACT

Businesses across all sectors and domains are combining their vast holdings of big data with high-powered analytics to find solutions to the complex and critical business problems in their industries. A challenge in moving to this data-driven paradigm is first incorporating the wide variety of data sources that companies have available. The health and life sciences industry is no different, as it encompasses everything from clinical trials, which include patient demographics and lab results, to insurance claims. SAS® Health: Data Mapper on SAS® Viya® provides a modern analytics and reporting platform for health-care data, and it gives you the power to achieve the following: define standards, using a flexible definition format, to explicitly outline the expected format of their final data structures; explore and visualize your data sources to better determine how to harmonize them with one another; map your data via a guided mapping process; export transformed data sets or the SAS® code generated by the mapping process; and take advantage of existing work to repeat the same transformations on different data sources. This paper includes a description of some of the common steps you need to perform during the transformation process, including data merging, matching, and ensuring quality. The examples provided are applicable to anyone who needs to transform data to a known model.

INTRODUCTION

Transforming data from one data model to another can be an inherently complex and tedious process further complicated by the following factors:

- the changing of the source or target data models
- the need to repeat the same transformations on similar, but "different enough" data sources
- the difficulty of balancing the traceability of the transformation being made to the source data with flexible and reusable transformation code
- verifying the integrity of transformed data

These challenges have given rise to an environment where the transformation of data sets is a highly manual process and results in limited repeatability, loss of productivity, and an increased chance of errors. SAS Health: Data Mapper is a new solution available from SAS to help life science organizations solve the complex data transformation problems. We held to the following tenets during the development of the solution:

1. An intuitive and guided mapping process: users should always be aware of their current state and what their next step should be.
2. Transparency and traceability: all changes made to the data sources throughout the mapping process should be transparent to the user. It should be traceable such that the user can both discover and make changes to mappings at a table and column granularity.
3. Repeatability: after a user has completed a mapping once, it should be possible to reuse the mapping with little or no changes necessary.

4. Cyclical: mapping data is not a linear process, so our solution should allow users to backtrack and make changes as needed as they make new discoveries.

This paper explores the features available in the Data Mapper and the process to transform raw data sources to a standardized format including defining data standards, the guided mapping process, and building mapping collections. To best describe the process, we will refer to an ongoing example throughout the paper where we work on mapping a set of clinical data sets. The data used is from the Nicardipine clinical trial (Haley et al., 1993) and we will map some of the data sets to the tables defined in the Study Data Tabulation Model (SDTM) provided by the Clinical Data Interchange Standards Consortium (CDISC). SDTM is the format expected for final clinical trials submitted to the U.S. Food and Drug Administration. Nicardipine is a medication used to treat high blood pressure and angina, and in this double-blinded study it was used to treat subarachnoid hemorrhage.

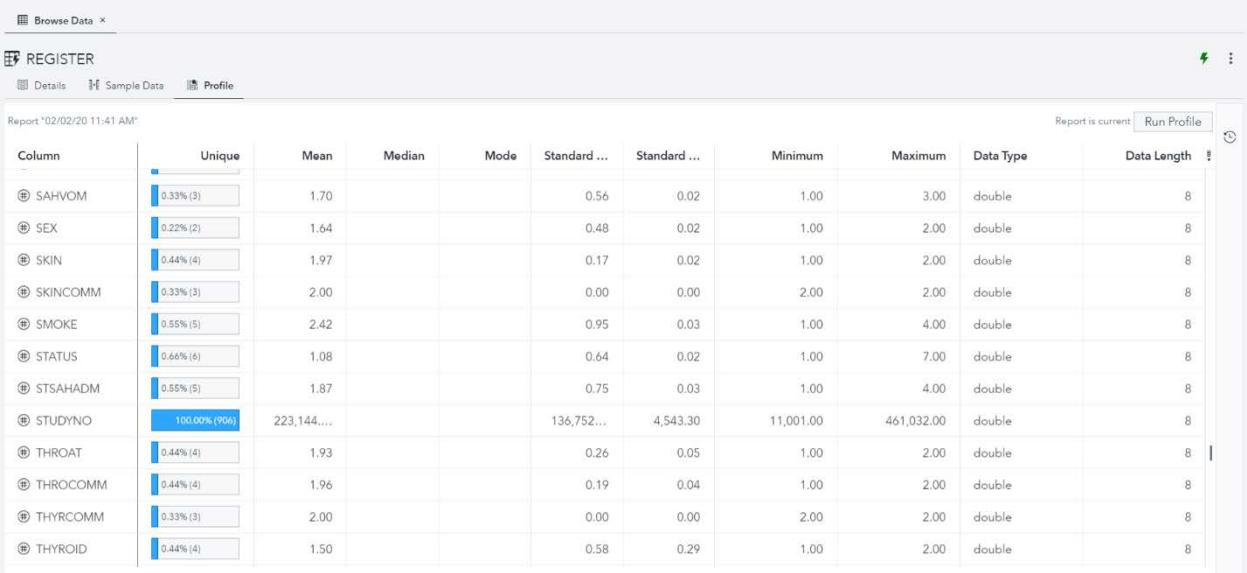
SAS Health: Data Mapper uses SAS Viya and SAS® Cloud Analytic Services (CAS) for all its data processing. As a result, the solution can use any data sources supported by the platform and its data connectors. This includes anything from path-based or server-side directories to Hadoop, AWS S3 or Redshift, and SQL or NoSQL databases. For users not familiar with SAS Viya, CAS, and caslibs, please see the recommended reading. Furthermore, users can see how the Data Mapper fits into the SAS platform and how it is used in conjunction with our available solutions such as SAS® Studio, SAS® Data Preparation, and SAS® Visual Analytics.

EXPLORE DATA

The first step for most users in the Data Mapper is data exploration, because it is crucial for users to understand how their data is organized and formatted before they can begin the mapping process. Users can use the data explorer to easily browse data and explore different aspects of a specific table including:

- Details: the columns in the table and their attributes
- Sample Data: a configurable display of the first 100 rows in the table to get an understanding of the types of values in each column
- Profile: a report to help identify anomalies or inconsistencies in the table using various descriptive statistics (see Display 1)

With a clinical trial there are usually metadata type files associated with a study including the protocol, annotated case report form (CRF), statistical analysis plan (SAP), and a data dictionary. All of these can help a user gain a better understanding of the data they have available, but this type of documentation is not always available for all data sources. As a result, the data explorer is a crucial tool for the user to understand the data that they will use in the mapping process.



Column	Unique	Mean	Median	Mode	Standard ...	Standard ...	Minimum	Maximum	Data Type	Data Length
SAHVOM	0.33% (3)	1.70			0.56	0.02	1.00	3.00	double	8
SEX	0.22% (2)	1.64			0.48	0.02	1.00	2.00	double	8
SKIN	0.44% (4)	1.97			0.17	0.02	1.00	2.00	double	8
SKINCOMM	0.33% (3)	2.00			0.00	0.00	2.00	2.00	double	8
SMOKE	0.55% (5)	2.42			0.95	0.03	1.00	4.00	double	8
STATUS	0.66% (6)	1.08			0.64	0.02	1.00	7.00	double	8
STSAHADM	0.55% (5)	1.87			0.75	0.03	1.00	4.00	double	8
STUDYNO	100.00% (906)	223,144....			136,752...	4,543.30	11,001.00	461,032.00	double	8
THROAT	0.44% (4)	1.93			0.26	0.05	1.00	2.00	double	8
THROCOMM	0.44% (4)	1.96			0.19	0.04	1.00	2.00	double	8
THYRCOMM	0.33% (3)	2.00			0.00	0.00	2.00	2.00	double	8
THYROID	0.44% (4)	1.50			0.58	0.29	1.00	2.00	double	8

Display 1. The Profile Tab of the Data Explorer Displaying the REGISTER Table

DEFINE DATA STANDARDS

The data explorer and its ability to browse data allows the user to explore the source data that is used in the mapping process, but we also need to define the target model before we can begin mapping. In the Data Mapper, we use data standards to define the tabular format – the tables and columns – of the target data model. A data standard can be based on existing industry recognized data models such as SDTM and OMOP (Observational Medical Outcomes Partnership), or it could be a data model used by your organization; data standards are flexible to your needs.

To add a data standard in the Data Mapper, a user must supply a name, and a description and version, to create it. More importantly, to populate the data standard, the user must upload a data standard definition file (described below). When a standard is initially created, its state is set to inactive, but for a standard to be used for mapping, it must be set to active. The different available state options for data standards enable users to indicate when a standard is ready for use, is under development, and when it has been retired from use. Once a definition file has been uploaded, the user can view the tables and columns of a standard within the application (see Display 3).

DATA STANDARD DEFINITION FILE

The data standard definition file is a UTF-8 encoded CSV file defining the tables and the columns in a data standard (see Display 2 for an example). The definition file has a flexible format such that non-required columns do not need to be included in the file and all other columns can appear in any order (see Table 1 for the required columns).

Column Name	Required	Description
Line_Identifier	True	A unique identifier for the row. This is useful when determining validation issues.
Table_Name	True	The name of a standard table.
Table_Label	False	A label of a standard table.
Table_Description	False	A description of a standard table.
Table_Constraint	True	Indicates whether a standard table is required, expected, or optional.

Column Name	Required	Description
Column_Name	True	The name of a standard column within a table.
Column_Label	False	A label of a standard column.
Column_Description	False	A description of a standard column.
Column_Type	True	Indicates the type of a standard column. It can be either character or numeric.
Column_Order	True	The order in which the standard column appears a standard table.
Column_Length	True	The length of the standard column.
Column_Format	False	Any format applied to the standard column.
Column_Constraint	True	Indicates whether a standard column is required, expected, or optional.

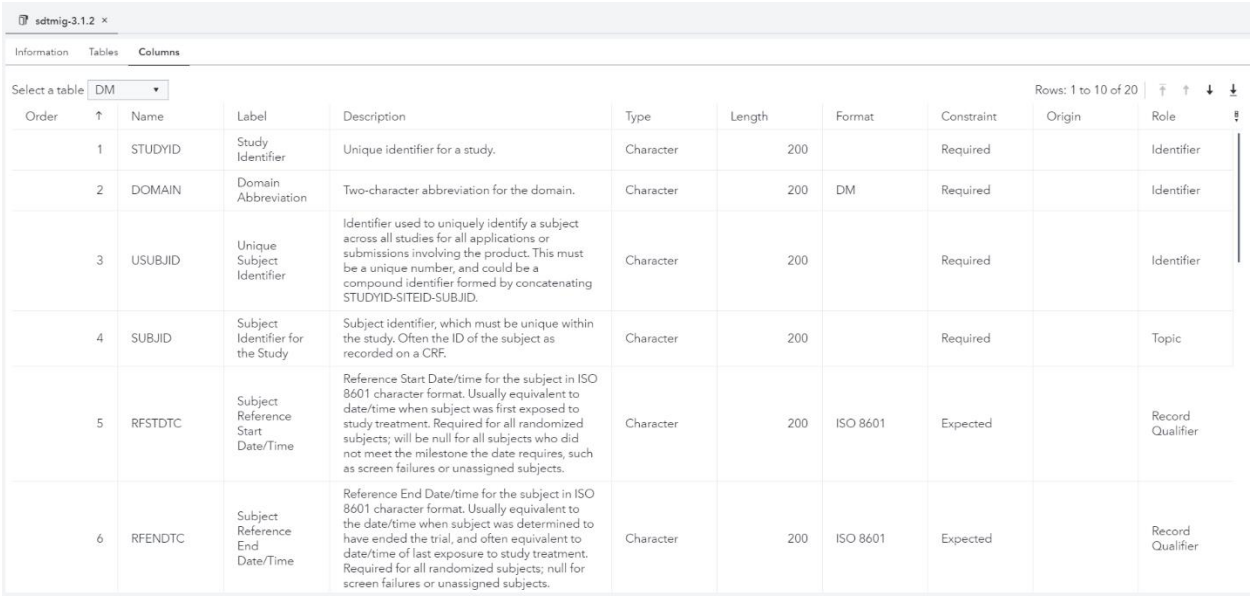
Table 1. Data Standard Definition File Columns

Furthermore, additional columns can be added to the data standard definition file to hold custom information. These are known as extended attributes and are easily added by prefixing the column with either 'Table_' or 'Column_' to indicate whether the attribute is related to the table or column in the data standard definition file.

We want to map the Nicardipine study to SDTM. To create an SDTM data standard, we downloaded the SDTMIG (SDTM Implementation Guide) 3.1.2 which is based on SDTM 1.4 from the CDISC website (note that an account is required) and then made a few alterations to match the expected format of the data standard definition file.

Line_Identifier	Table_Name	Table_Label	Table_Constraint	Table_Keys	Column_Name	Column_Label	Column_Description	Column_Type	Column_Order
1	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	STUDYID	Study Identifier	Unique identifier for a study.	1	
2	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	DOMAIN	Domain Abbreviation	Two-character abbreviation for th	1	
3	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	USUBJID	Unique Subject Identifier	Identifier used to uniquely identif	1	
4	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AESEQ	Sequence Number	Sequence Number given to ensur	0	
5	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEGRPID	Group ID	Used to tie together a block of re	1	
6	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEREFID	Reference ID	Internal or external identifier such	1	
7	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AESPID	Sponsor-Defined Identifier	Sponsor-defined identifier. It may	1	
8	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AETERM	Reported Term for Adverse Event	Verbatim name of the event.	1	
9	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEMODIFY	Modified Reported Term	If AETERM is modified to facilitate	1	
10	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEDECOD	Dictionary-Derived Term	Dictionary-derived text descriptio	1	
11	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AECAT	Category for Adverse Event	Used to define a category of relat	1	
12	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AESCAT	Subcategory for Adverse Event	A further categorization of advers	1	
13	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEPRESP	Pre-specified Adverse Event	A value of 'Y' indicates that this a	1	
14	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEBODSYS	Body System or Organ Class	Body system or organ class used	1	
15	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AELOC	Location of the Reaction	Describes anatomical location rel	1	
16	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AESEV	Severity/Intensity	The severity or intensity of the ev	1	
17	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AESER	Serious Event	Is this is a serious event?	1	
18	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEACN	Action Taken with Study Treatmer	Describes changes to the study tr	1	
19	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEACNOTH	Other Action Taken	Describes other actions taken as	1	
20	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEREL	Causality	Records the investigator's opinior	1	
21	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AERELNST	Relationship to Non-Study Treatm	Records the investigator's opinior	1	
22	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEPATT	Pattern of Adverse Event	Used to indicate the pattern of th	1	
23	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AEOUT	Outcome of Adverse Event	Description of the outcome of an	1	
24	AE	Adverse Events	0	STUDYID, USUBJID, AEDECOD, A	AESCAN	Involves Cancer	Was the serious event associated	1	

Display 2. The Data Standard Definition File for SDTMIG 3.1.2

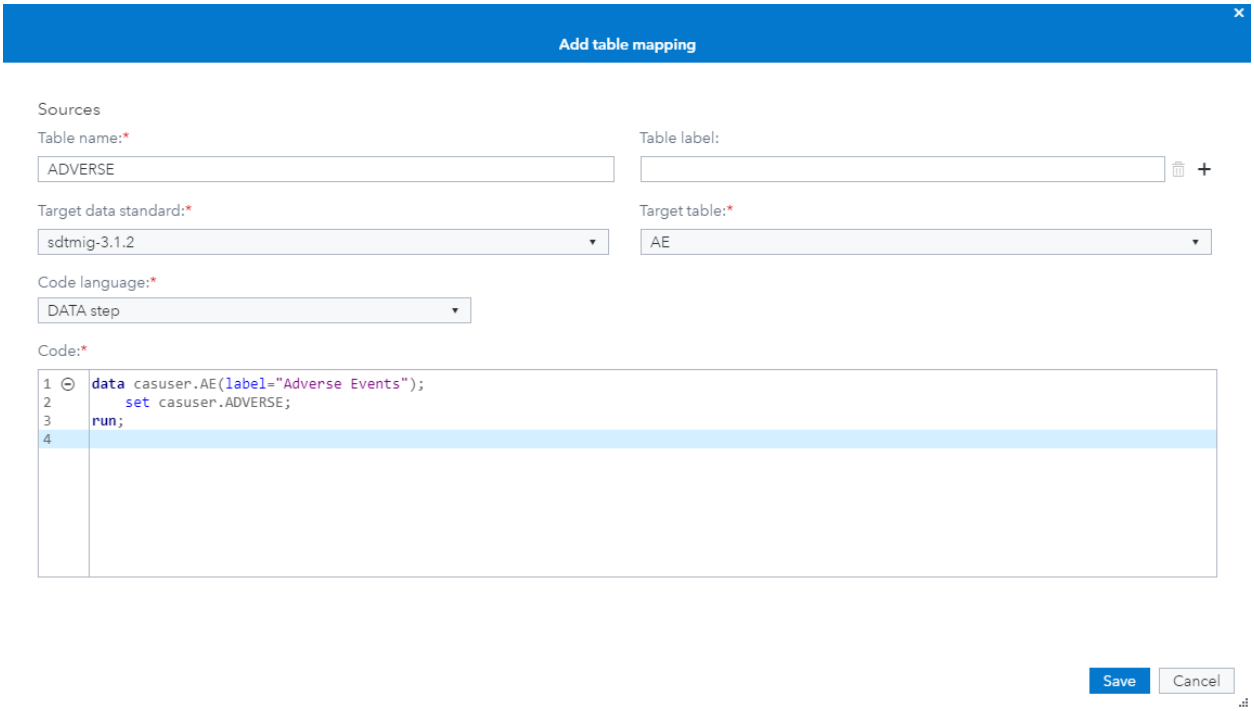


Order	Name	Label	Description	Type	Length	Format	Constraint	Origin	Role
1	STUDYID	Study Identifier	Unique identifier for a study.	Character	200		Required		Identifier
2	DOMAIN	Domain Abbreviation	Two-character abbreviation for the domain.	Character	200	DM	Required		Identifier
3	USUBJID	Unique Subject Identifier	Identifier used to uniquely identify a subject across all studies for all applications or submissions involving the product. This must be a unique number, and could be a compound identifier formed by concatenating STUDYID-SITEID-SUBJID.	Character	200		Required		Identifier
4	SUBJID	Subject Identifier for the Study	Subject identifier, which must be unique within the study. Often the ID of the subject as recorded on a CRF.	Character	200		Required		Topic
5	RFSTDTC	Subject Reference Start Date/Time	Reference Start Date/time for the subject in ISO 8601 character format. Usually equivalent to date/time when subject was first exposed to study treatment. Required for all randomized subjects; will be null for all subjects who did not meet the milestone the date requires, such as screen failures or unassigned subjects.	Character	200	ISO 8601	Expected		Record Qualifier
6	RFENDTC	Subject Reference End Date/Time	Reference End Date/time for the subject in ISO 8601 character format. Usually equivalent to the date/time when subject was determined to have ended the trial, and often equivalent to date/time of last exposure to study treatment. Required for all randomized subjects; null for screen failures or unassigned subjects.	Character	200	ISO 8601	Expected		Record Qualifier

Display 3. The Columns in the DM Table for the SDTMIG 3.1.2 Data Standard

CREATE MAPPING COLLECTIONS

A mapping collection is a resource intended to contain curated lists of table and column mappings. The purpose of storing individual table and column mappings is so that the user can reuse successful mappings already developed. When manually mapping raw data sources to a target data standard, a user often performs the same types of transformation over and over. When these mappings are stored in a mapping collection, the user can use the Automap feature to easily reapply the mappings in new mapping projects with little to no manual changes necessary. Table and column mappings can be added to a mapping collection manually (see Display 4) or exported in the Export step of a mapping project. Mapping collections are globally available to all users, so users can collaboratively develop and manage mapping collections to benefit an entire organization.



Display 4. Manually Creating a Table Mapping In a Mapping Collection

MAP DATA

After exploring the Nicardipine study with the data explorer and defining SDTMIG 3.1.2 as a data standard, we can begin mapping data. The entire mapping process is encapsulated in a mapping project, which is a self-contained resource for the mapping of some array of raw data sources to a specific data standard. Unlike data standards or mapping collections, mapping projects are stored as SAS content and accessible via SAS Drive. This enables users to organize projects however they like and freely define the authorization on a mapping project to control access.

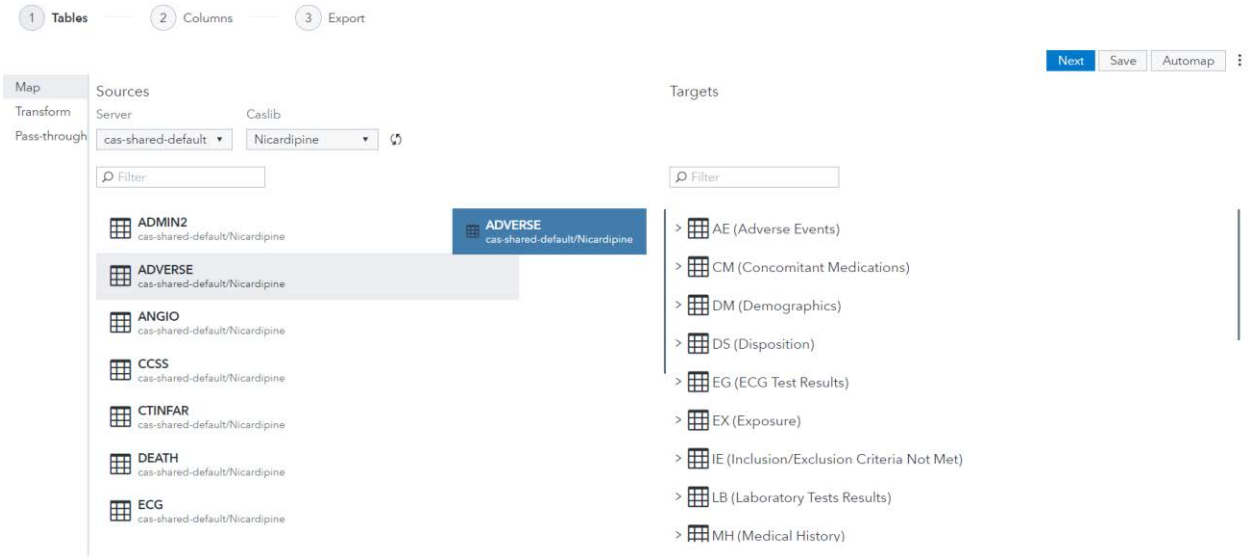
When creating a mapping project, the user must define a name, a data standard, and a code language for the project. The user can choose from any active data standard and select either DATA step or CASL as the programming language for the project. Although the user is free to use Data Step, CASL, macros, and even DS2 within the mapping project, only the selected programming language is used for the autogenerated transformation code. Furthermore, the data standard and programming language cannot be changed after a mapping project is created. To learn more about CASL, please see the Recommended Reading section.

The user can browse data sources, view data standards, and have multiple mapping projects open in separate tabs allowing for easy cross-referencing and comparison throughout the mapping process. Mapping projects are isolated from each other because all transformations are executed in separate SAS compute sessions, so interference is not an issue when different users are working or the same data sources are being used across mapping projects.

The mapping project contains three steps that the user can freely navigate between at any time: Table, Column, and Export. First, the user maps source data tables to their appropriate target data standard tables. Second, the user maps the source columns within each table to their corresponding target columns. Lastly, the user can choose to export the mapping project code, the mapped tables, or the mappings created in the project. We will explore each of these steps throughout the rest of the paper.

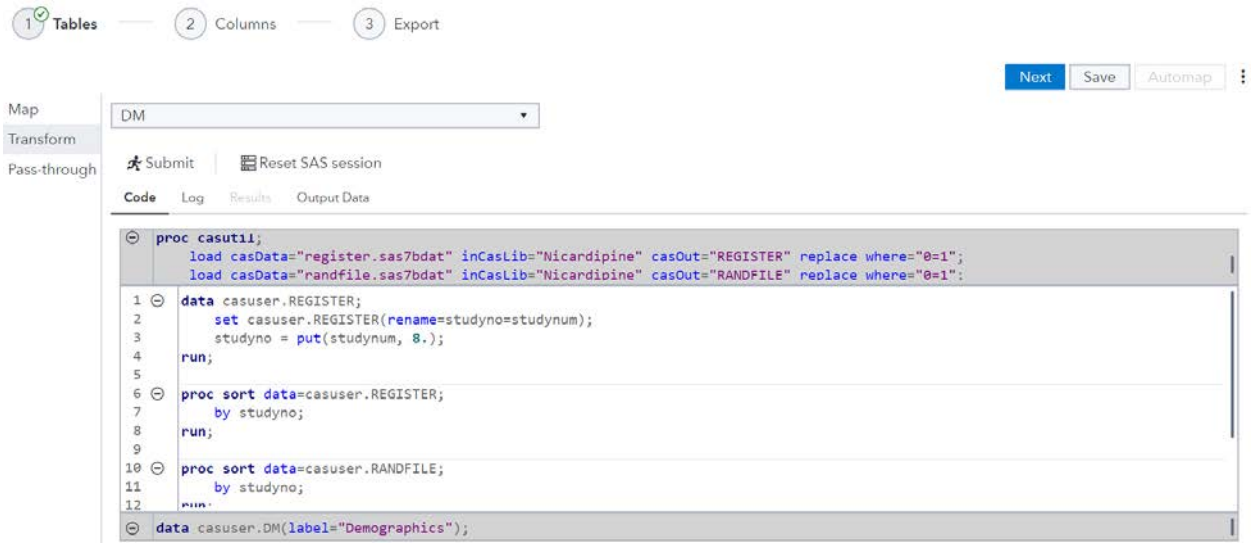
TABLES

In the Tables step, the user maps the data tables from any data source available on the platform to the appropriate target table in the selected data standard. The user can perform both one-to-one or many-to-one mappings of source table(s) to a target table using an easy drag and drop interface (see Display 5). Users can remove sources individually from a table mapping at any time.



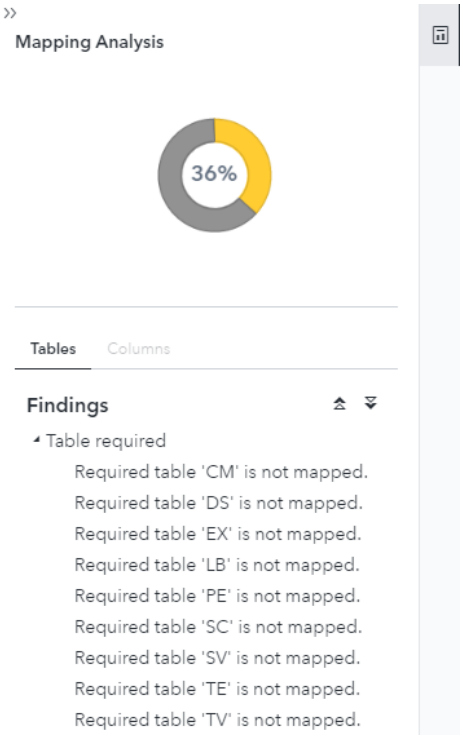
Display 5. Mapping ADVERSE to AE with Drag and Drop Interface

After a mapping is completed, transformation code is autogenerated for the user and is available in the Transform sub-tab (see Display 6). For a one-to-one mapping, a copy operation is performed. For a many-to-one mapping of sources, a union is performed. If the user needs to provide more complex transformation code such as redefining variable names, converting a variable's type, or performing a table merge, this can all be accomplished in the SAS code editor that is provided. A separate code editor is available for each standard table, which provides greater traceability of the transformations performed. All the autogenerated transformation code provided for the user is executed within zero observation CAS tables to provide a highly performant environment during the mapping process. Only during the final Export step are the transformations performed on the entirety of the source data tables.



Display 6. Table Transform Code Editor for DM (Demographics) Table

Each time the user saves to their mapping project, a mapping analysis is executed (see Display 7). The mapping analysis indicates to the user whether any required tables, as determined by the selected data standard, are unmapped or have missing transformation code. The mapping analysis includes a gauge to provide a helpful indicator of mapping progress. Progress is determined based on how many of the required or expected tables and columns in the data standard have been mapped.

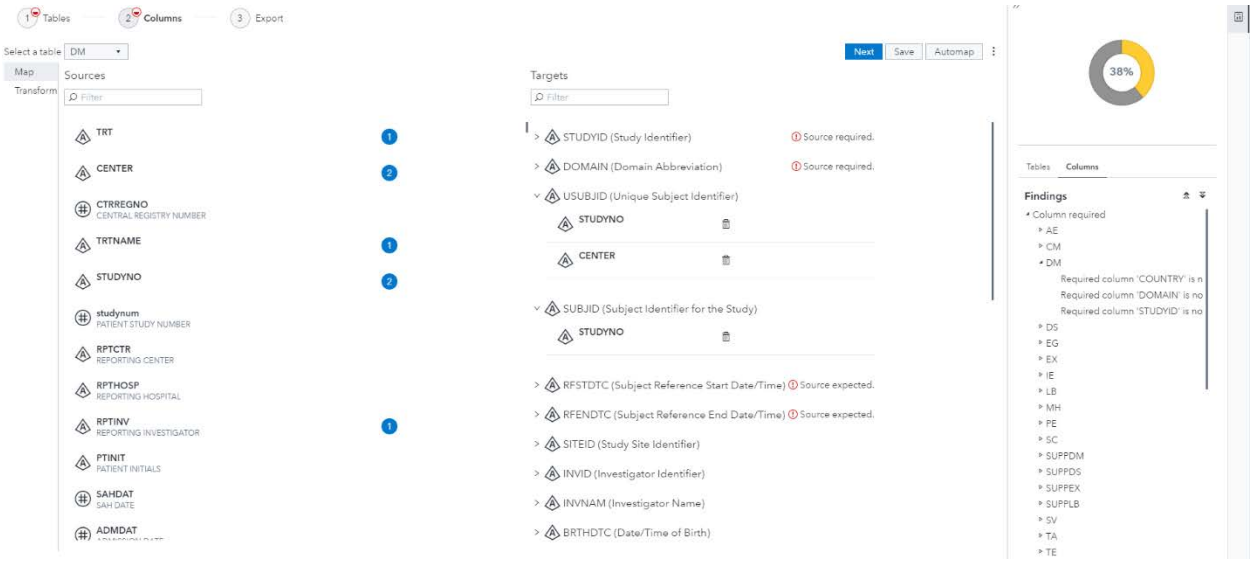


Display 7. Mapping Analysis of Table Mappings

Lastly, in the Pass-through sub-tab the user has the option to select data tables available on the platform to be available during the final Export step. This enables the user to co-locate any tables not used during the mapping process with the transformed target tables.

COLUMNS

After a table has been mapped or transformation code has been provided, the user can begin mapping the columns of a standard table in the Columns step. The source columns are the columns from the table(s) mapped in the previous step and the target columns are those defined in the data standard. The user can perform column mappings with a drag-and-drop interface; this includes both one-to-one and many-to-one mappings (see Display 8). As in the Table step, the user can remove a source column from a mapping at any time and can view a mapping analysis. The mapping analysis indicates to the user which required or expected columns are unmapped or are missing transformation code.

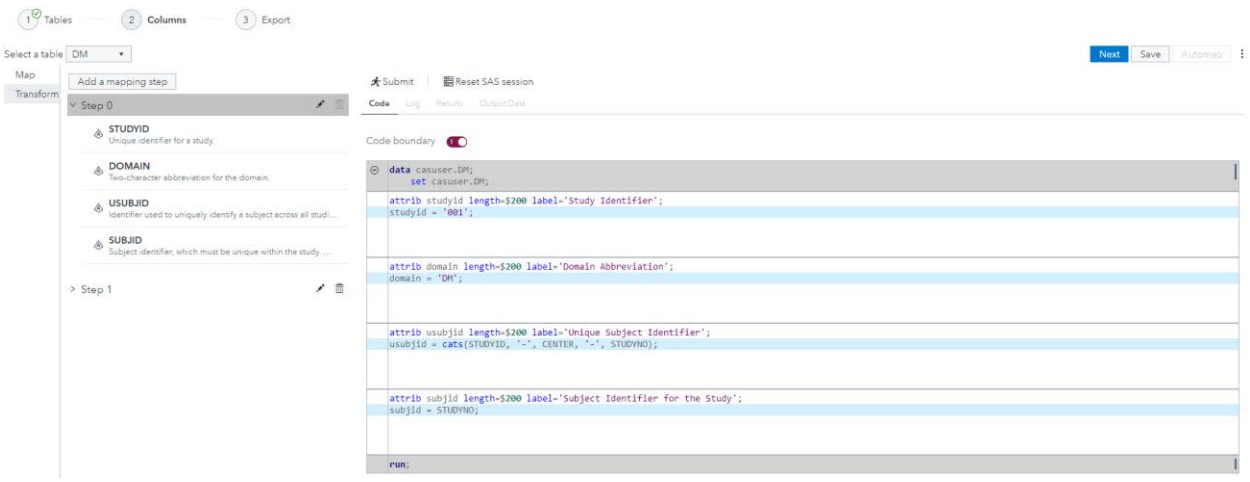


Display 8. Column Mapping for standard table DM with Mapping Analysis

The Transform sub-tab is different in the Columns step because order of execution matters greatly when mapping columns. Often columns within a table are derived or calculated from other columns in the same table; if columns are not mapped and assigned values in the proper order, then errors can occur, or incorrect data is assigned to derived columns.

To ensure that column mappings execute in a well-defined order, mapping steps are provided (see Display 9). Each step and all the column mappings within a step execute in a linear order. Furthermore, a mapping step acts as a step boundary, which is helpful for organizing the mapping of columns into manageable blocks, but also can be necessary from a programmatic standpoint. For example, if the user would like to use a procedure or do any post-processing with DATA step, they can create a new mapping step and toggle code boundary. When code boundary is switched off, no step boundary is automatically provided for the user (either a DATA step or PROC CAS) and the user has complete freedom to write any transformation code necessary.

Just as in the Tables step, each standard column is provided with a separate code editor. This provides traceability at a column-level granularity and provides two important features. One, it is helpful from a data lineage or auditing perspective in determining how tables and columns are changed throughout the mapping process. Two, it is crucial for encapsulating the transformations necessary for mapping an individual column and this comes into fruition when the user exports table and column mappings in the final Export step.



Display 9. Column Transform for Standard Table DM with Multiple Mapping Steps

EXPORT

Within the final Export step, the user has a few different options available depending on their goal in mapping their data.

Generate Project Code

The user can choose to export the generated transformation code created throughout the mapping process to continue their work in SAS Studio or SAS Data Preparation. Furthermore, they can export the code and set up a job within SAS® Environment Manager to execute the transformation on a schedule.

There is the option to export the generated transformation code to either a ZIP file or to SAS Drive depending on whether the user plans to continue work within the SAS platform or outside of it (see Display 10). A separate SAS program file is generated for each table defined in the selected data standard for the mapping project. In addition, the user has the option to generate an audit report that provides a snapshot of the current state of the mapping project.

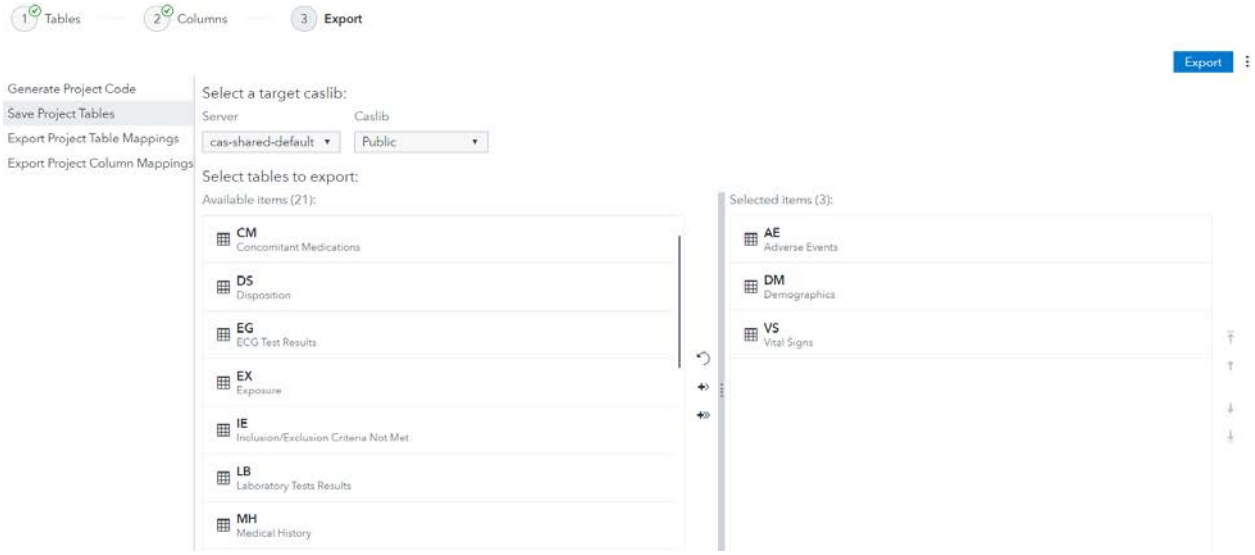
Display 10. Generating Mapping Project Transformation Code

Save Project Tables

Alternatively, the user might intend to use their mapped data in a visualization or a machine learning model. If so, they can export the transformed tables to a global caslib and use those tables within SAS Visual Analytics, SAS® Visual Data Mining and Machine Learning, or any other SAS product on the platform.

The user can select the final target tables, defined by the data standard, and any pass-through tables that they have selected in the Tables step (see Display 11). Throughout the mapping process all the transformations have been executing against zero observations tables for efficiency, but when saving the project tables the transformation code is executed

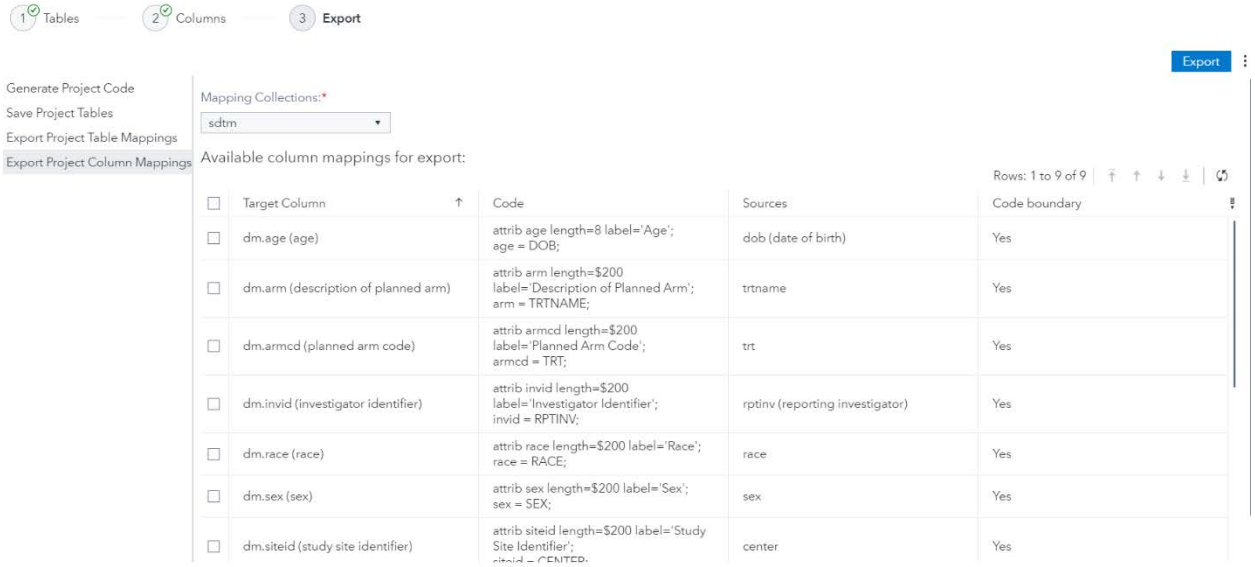
on the full data sets. As a result, saving project tables can depend on the size of source data tables. The user receives a notification when the export has started and when it has completed. This enables the user to continue working while the saving of the project tables is executing.



Display 11. Saving Standard Tables AE, DM, and VS from Mapping Project to Public Caslib

Export Project Table and Column Mappings

Finally, the user can export table and column mappings to a predefined mapping collection which has potential to be reused (see Display 12). Only unique table and column mappings that do not exist in the selected mapping collection are shown with the option to export. By exporting mappings to a mapping collection, the user can reduce the amount of time it takes to do similar transformations in the future.



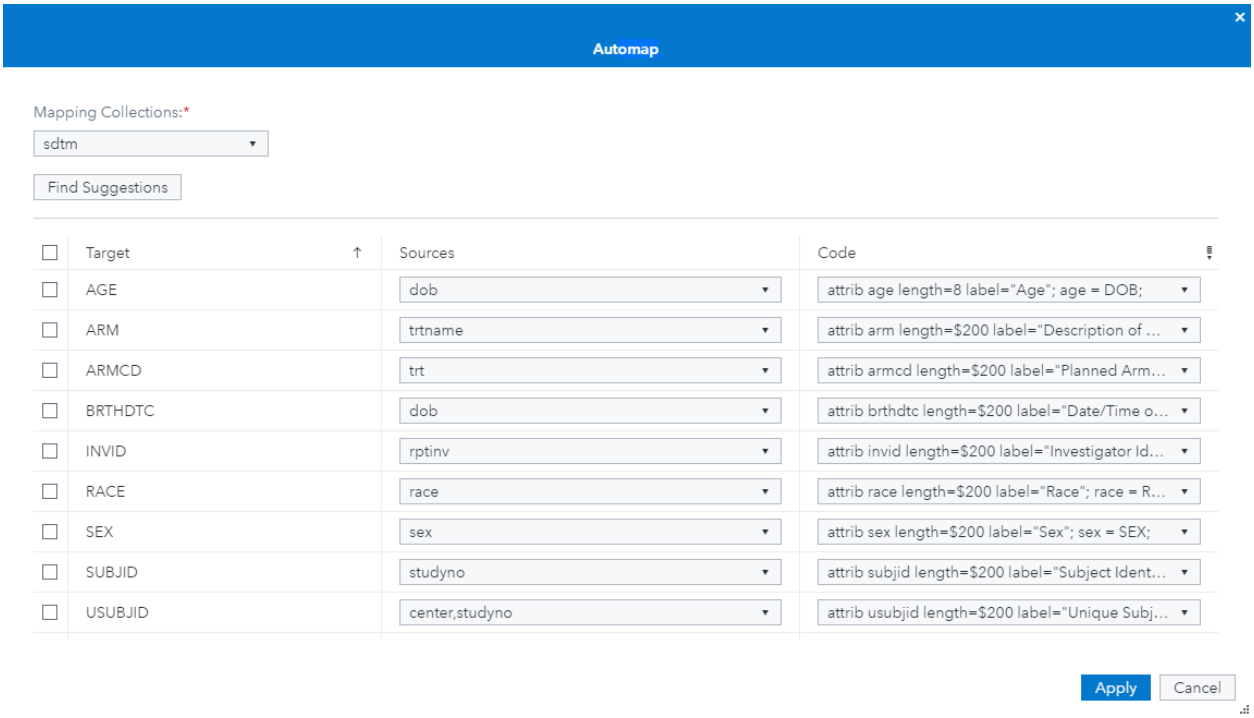
Display 12. Exporting Mapping Project Column Mappings to a Mapping Collection

AUTOMAP

The Automap feature is available in the Map sub-tab of the Tables and Columns steps in a mapping project. With Automap, users can reuse relevant and repeatable mapping across

mapping projects that have been stored in a mapping collection. This reduces the time it takes the user to map data sources to a data standard. In the table Automap, the user must select a source caslib and mapping collection to provide suggestions from, but in the column Automap the user need only select a mapping collection. The more the mappings in a mapping collection are representative of the raw data sources, the better the suggestions provided will be.

In the Automap dialog box, the user can select any number of suggestions to apply to their mapping project, but suggestions are displayed only for unmapped tables or columns (see Display 13). After the user has applied suggestions, they can modify the mappings in the map and transform sub-tabs. Suggestions are organized by target data standard tables or columns and both one-to-one and many-to-one mappings can be suggested.



Display 13. Column Automap for Standard Table DM

CONCLUSION

SAS Health: Data Mapper helps users solve the complex data transformations in their organizations. The solution gives the user an environment where the entire life cycle of data transformation can be managed, specifically including the following abilities:

- explore data sources available on the SAS platform
- define and manage data standards
- map and export data through a guide process
- automap common and reusable mappings

For organizations with large, complex, and disparate data sources, SAS Health: Data Mapper provides a point-and-click interface to manage the integration of data into new, variable structures for analysis and reporting. Thus, helping users to turn data into intelligence.

REFERENCES

Haley, EC, et al. 1993. " A randomized controlled trial of high-dose intravenous nicardipine in aneurysmal subarachnoid hemorrhage: A report of the Cooperative Aneurysm Study." *Journal of Neurosurgery*, 78: 537-547. Available https://doi.org/10.3171/jns.1993.78.4.0537.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to the SAS Health: Data Mapper development team for their tireless work for which without this paper would not be possible, especially Nathan Asselstine and Rick Wachowiak.

I would also like to thank the following people for their feedback and thoughts during the paper writing process: Brian Durham, Elliot Inman, Gene Lightfoot, Rick Monaco, and Heather Weinstein.

RECOMMENDED READING

- Pendergrass, Jerry. 2017. "The Architecture of the SAS® Cloud Analytic Services in SAS® Viya™." *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available https://support.sas.com/resources/papers/proceedings17/SAS0309-2017.pdf.
- SAS Institute Inc. 2020. An Introduction to SAS® Viya® 3.5 Programming. Cary, NC: SAS Institute Inc. Available https://documentation.sas.com/?docsetId=pgmdiff&docsetTarget=titlepage.htm&docsetVersion=3.5.
- SAS Institute Inc. 2019. SAS® Cloud Analytic Services 3.5: CASL Programmer's Guide. Cary, NC: SAS Institute Inc. Available https://documentation.sas.com/?docsetId=caslpg&docsetTarget=titlepage.htm&docsetVersion=3.5.
- SAS Institute Inc. 2019. SAS® Cloud Analytic Services 3.5: CASL Reference. Cary, NC: SAS Institute Inc. Available https://documentation.sas.com/?docsetId=proccas&docsetTarget=titlepage.htm&docsetVersion=3.5.
- SAS Institute Inc. 2019. SAS® Cloud Analytic Services 3.5: DATA Step Programming. Cary, NC: SAS Institute Inc. Available https://documentation.sas.com/?docsetId=casdspgm&docsetTarget=titlepage.htm&docsetVersion=3.5.
- SAS Institute Inc. 2019. SAS® Cloud Analytic Services 3.5: User's Guide. Cary, NC: SAS Institute Inc. Available https://documentation.sas.com/?docsetId=casref&docsetTarget=titlepage.htm&docsetVersion=3.5.
- SAS Institute Inc. 2019. SAS® Data Explorer 2.2: User's Guide. Cary, NC: SAS Institute Inc. Available https://documentation.sas.com/?docsetId=datahub&docsetTarget=titlepage.htm&docsetVersion=2.2.
- Shannon, David. 2018. "Come On, Baby, Light my SAS® Viya®: Programming for CAS." *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc. Available https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2622-2018.pdf.

- Sober, Steven. 2019. "What is CASL?" *Proceedings of the SAS Global Forum 2019 Conference*. Cary, NC: SAS Institute Inc. Available https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3040-2019.pdf.

CONTACT INFORMATION

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

Eric Bolender
SAS Institute
Eric.Bolender@sas.com