

Managing Custom Data Standards in SAS Clinical Data Integration

Melissa R. Martinez, SAS Institute, Inc., Round Rock, Texas, United States

ABSTRACT

SAS® Clinical Data Integration (CDI) is built on the idea of transforming source data into CDISC standard domains such as SDTM and ADaM. However, you can also define custom data standards to use within CDI. This paper describes several scenarios for using custom data standards, such as incorporating company-specific requirements, developing therapeutic area or compound level domains, and even using study-specific data standards. This paper also describes in detail how to set up the required data sets to define a custom data standard, register the custom data standard with SAS Clinical Standards Toolkit (CST), and import the custom data standard into SAS Clinical Data Integration. In addition, best practices are discussed for both managing the security of SAS Clinical Standards Toolkit Global Library that is central to the SAS Clinical Data Integration application and for the overall process of developing custom data standards.

OVERVIEW OF USING DATA STANDARDS IN SAS CLINICAL DATA INTEGRATION

There are numerous ways SAS Clinical Data Integration helps users implement CDISC data standards. SAS Clinical Data Integration is built using SAS Data Integration Studio as its foundation. Then SAS Clinical Standards Toolkit is integrated into it, which provides metadata about the CDISC data standards and controlled terminology, as well as tools to check compliance of study domains to the data standard.

Within the user interface of SAS Clinical Data Integration, users can import data standards. These data standards come directly from SAS Clinical Standards Toolkit. There are several versions of SDTM, ADaM, and SEND data standards available for import. A data standard that has been imported into SAS Clinical Data Integration contains domain templates, which contain all of the metadata about each domain. This includes all of the columns possible for the domain and their label, length, and format. Each column also has its own metadata; such as whether it is Required, Expected, or Permissible, whether it is a key variable, its XML data type, and any associated controlled terminology codelist. Each domain also has metadata defined. This includes its structure, title and file name of its archive file (SAS v5 transport file), and its key variables.

When a user creates a new Study in SAS Clinical Data Integration, they choose which data standard(s) they would like associated with the Study. Then, the user can create new Standard Domains within the study, where they choose which domains from the associated data standard(s) they would like to create. At the time they are created, these domains are metadata objects within SAS Clinical Data Integration and are an exact copy, including all metadata for the domains and their columns, of the domain templates from the data standard. From there, users create SAS Clinical Data Integration jobs to populate those domain instances with data by transforming the source data into the structure required by the domains.

Another piece of the submission puzzle is the define.xml document that accompanies the submission. This document describes the study, all of the data sets being submitted, their structure, codelists used, computational algorithms, comments, value level metadata, and more. SAS Clinical Data Integration has a transformation that can create the define.xml using the metadata from the study, its domains, and the Controlled Terminology Package associated with the study.

LIMITATIONS OF GENERAL CDISC DATA STANDARDS

Most pharmaceutical companies have adopted the CDISC SDTM and ADaM data standards for submission of clinical study data to regulatory agencies. While these standards have come a long way in providing a standard format for submission data, the CDISC data models as they are published do not typically fit the data for a given clinical study perfectly.

For example, in the SDTM model each column within a domain is given a designation by CDISC as Required, Expected, or Permissible. CDISC provides a way to report much of the data that it finds to be commonly collected, with the understanding that some things may not apply to all studies. So, for any given study, it is likely that a company would remove some of the Permissible and/or Expected columns from their domains.

Many companies already had some kind of internal data standards implemented before CDISC started releasing their data standards. It is common for companies to use some hybrid of the CDISC standards and their internal, company-specific data standards.

Most companies also find that the data they have collected for their studies does not fit perfectly into the existing CDISC data models. In this case, the company will often need to create additional variables within existing domains or create new custom domains altogether.

Different therapeutic areas often have unique data collected. While CDISC has an initiative to develop Therapeutic Area data standards, these are still in development. Many companies have established their own standard domain templates for therapeutic area-specific data they are collecting.

PhUSE 2014

In the define.xml file created by the SAS Clinical Data Integration transformation, the source, algorithm, and comment values are not populated for domain columns by default. These are values that cannot be anticipated in a way that would allow standard values to be set by CDISC or SAS, and they are likely to be different in every study.

All of the study- or organization-specific items discussed above can be added or managed within the SAS Clinical Data Integration user interface. A user can add column-level clinical metadata within a study by opening the Properties of a domain, then opening the Properties of the column, and modifying the values on the Clinical tab. Source, algorithm, and comments can be added here. Custom domains, such as a therapeutic area-specific domain, can be created within a study and then promoted to the data standard so that it will be available to future studies in SAS Clinical Data Integration. Columns can be added or removed from domains within a study and their properties, such as label and length, can be modified. The domain templates within data standards themselves can be modified to add or remove columns and change clinical metadata about the domain and/or its columns. This means that each study using the data standard would use this modified metadata for its domains.

However, all of this modification and management of metadata within the user interface is time-consuming! Imagine adding the source and algorithm for every domain column within a clinical study this way.

THE BEST OF BOTH WORLDS: CUSTOM DATA STANDARDS BASED ON CDISC DATA STANDARDS

There is a better way to manage custom data standard needs. Using the existing standards in SAS Clinical Standards Toolkit as a template, create all of the data sets and files needed for a custom data standard and register it to SAS Clinical Standards Toolkit. Once a data standard has been properly registered to SAS Clinical Standards Toolkit, it will be visible to SAS Clinical Data Integration. The data standard can then be imported into SAS Clinical Data Integration and used for Studies.

Going a step further, there is a solid case to be made for using even study-specific custom data standards in SAS Clinical Data Integration. During the typical setup of a clinical study's programming effort, a programming specification document is produced that describes the submission domains that need to be created. This usually includes metadata about the target domain columns (label, length, etc.). The domains needed are specified, and directions for mapping or deriving columns are provided. The person developing these specifications usually has intimate knowledge of the source data and the study itself. This makes it straightforward to determine the source for each column. All of this information relates very closely to the metadata used in SAS Clinical Data Integration for clinical domains, but with the added benefit of being able to define additional column-level metadata. With this metadata defined already, it can be used to create a more complete define.xml file.

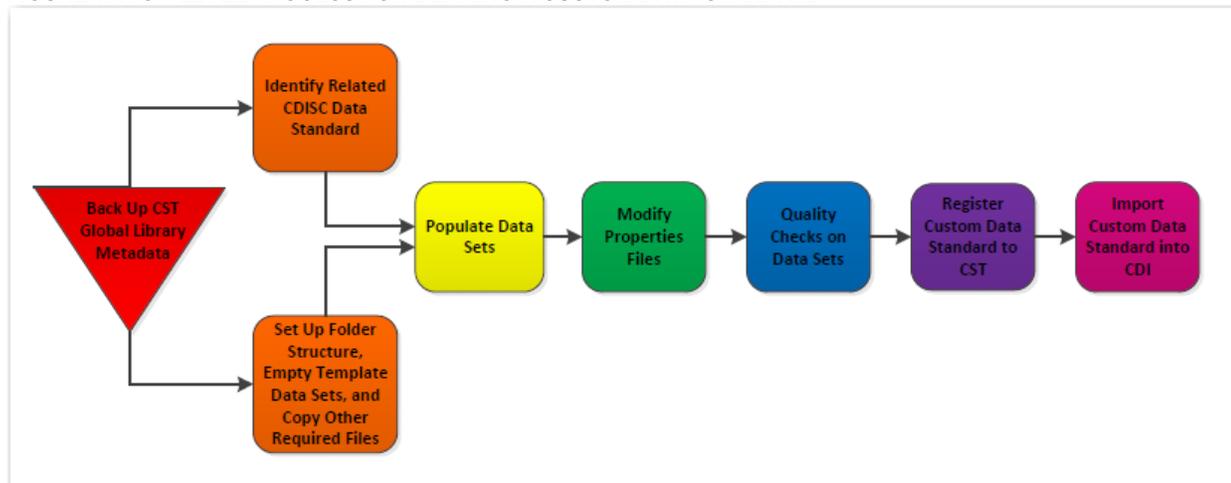
THE HOW-TO: CREATE YOUR OWN CUSTOM DATA STANDARD

The process for defining a custom data standard and registering it to SAS Clinical Standards Toolkit is an advanced topic, but it can fit very nicely into the existing process for creating study programming specifications for populating submission data sets. The next section of this paper will describe in detail the structure of the folders and data sets needed for a custom data standard, how to populate them most effectively, and how to register the data standard to SAS Clinical Standards Toolkit. Custom data standards can be used to define company-wide standards that are based on a CDISC data standard, therapeutic area-specific data standards, and study-specific data standards. Within SAS Clinical Data Integration, custom data standards should be based on a supported CDISC data standard.

OVERVIEW

First, let's take a look at the high-level process for defining and registering a custom data standard (illustrated in Figure 1). There are 13 SAS data sets that together provide all of the metadata for a data standard. They are stored in a standard folder structure that must be used for custom data standards. These data sets must all be created and populated, and the most straightforward way to do this is to copy as much as possible from an existing data standard.

FIGURE 1: HIGH-LEVEL PROCESS FOR DEFINING A CUSTOM DATA STANDARD



PhUSE 2014

The first step in getting started is to back up the SAS Clinical Standards Toolkit Global Library metadata. This library is the heart of SAS Clinical Data Integration, and great care should be taken with its contents. Next, identify the CDISC data standard on which your custom data standard is based, and also set up the folder structure and the 13 empty template data sets. There are also several SAS macros and other files to copy over from the related CDISC data standard. Next, the data sets for the custom data standard need to be populated. Many of them can have their contents copied over from the related CDISC data standard with little or no modification. There are two properties files for the data standard. One of these files requires updating and the other should be reviewed. Once all of the data sets and files have been created and/or modified, quality checks should be performed on the data sets to be sure they are populated as expected, required values are non-missing, and certain key values like the data standard version are consistent throughout all of the data sets. Then, the data standard is registered to SAS Clinical Standards Toolkit, which means that the metadata about the standard from your custom data standard's control data sets is added to the master SAS Clinical Standards Toolkit data standard metadata data sets. From there, the data standard can be imported into SAS Clinical Data Integration.

In the following sections each of these steps is described in detail using a study-specific custom data standard as an example. The study's title is *Blinded Controlled Efficacy Study of Nicardipine in Patients with Subarachnoid Hemorrhage – Nicardipine 0.75 mg vs. Nicardipine 0.15 mg*. The study's short name or abbreviation is NICSAH-01, which is how it will be referenced from here forward.

BACK UP CST GLOBAL LIBRARY METADATA

The SAS Clinical Standards Toolkit Global Library is typically a folder named `cstGlobalLibrary`, although it can vary from system to system. Under the parent folder are several subfolders, which may vary some depending on the version of SAS Clinical Standards Toolkit, although all versions contain a metadata subfolder and a standards subfolder. The metadata subfolder contains data sets which contain the high-level metadata for all of the data standards registered to SAS Clinical Standards Toolkit. For this reason, in this paper these data sets are referred to as the master metadata data sets. SAS Clinical Data Integration depends on information from these data sets at various times, such as when a user imports data standards. At this time CDI obtains the metadata about the available data standards from these data sets.

The standards folder contains the specific metadata for each of the included data standards. Each data standard, as well as the general SAS Clinical Standards Toolkit framework, has its own folder beneath the standards folder of the CST Global Library.

Before beginning the process of creating a custom data standard, it is important to back up the CST Global Library. At a minimum SAS recommends making a backup copy of the current metadata folder. The user may also choose to back up the entire CST Global Library. Another option is to use version control software to place the files in the CST Global Library under version control so that they can be reverted to earlier versions at any time. Regardless of the method chosen, it is imperative to make some kind of backup before proceeding, because the CST Global Library is central to the function of SAS Clinical Data Integration.

IDENTIFY RELATED CDISC DATA STANDARD

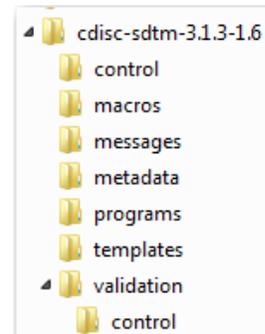
Custom data standards should be based on an existing CDISC data standard supported by your version of SAS Clinical Data Integration. Identify the CDISC data standard your custom data standard is based on. Throughout the rest of these instructions, this will be referred to as the reference data standard. For the NICSAH-01 study, CDISC SDTM 3.1.3 will be the reference data standard.

The reference data standard is important because it is far more straightforward to populate the 13 required data sets by copying much of the contents of these data sets from the reference data standard. In fact, for a typical study-specific data standard, most of these data sets can be copied from the reference data standard with either no modification or by just updating the data standard version. There are also some other files and data sets that should be copied from the reference data standard, which will be covered in the next section.

SET UP FOLDER STRUCTURE, EMPTY TEMPLATE DATA SETS, AND COPY OTHER REQUIRED FILES

There is a standard folder structure for the data sets and files that define a data standard that should be used. When setting up a new data standard, either create the folder structure from scratch or copy the folder structure from the reference data standard. Figure 2 shows the CDISC SDTM 3.1.3 data standard's parent folder along with the required subfolders.

FIGURE 2: FOLDER STRUCTURE



The name of the parent folder should be a valid folder or directory name for the user's operating system and should clearly describe which data standard it contains. For the example NICSAH-01 study-specific data standard, the folder will be named `study-nicardipine-sah-01-v1.0` (the prefix of "study" is an indicator that it is a study-specific data standard, a naming convention that can be convenient when the number of custom data standards becomes large). The parent folder will then be placed in the standards folder in the CST Global Library along with all of the existing CDISC data standards. A custom data standard's folder may be located anywhere on the user's file system, but care must be taken to set the file paths correctly in the data sets that describe the standard and the location of its related files.

PhUSE 2014

When copying the reference data standard's entire folder structure along with the files contained in it, as a best practice SAS recommends deleting the 13 data sets that you will populate with your custom data standard data. These data sets are in the control, messages, metadata, and validation/control folders. There will be several files, programs, and data sets in the macros, programs and templates folders, which should remain.

If the user chooses to create the folder structure from scratch, there will be only empty folders. Copy the files from your reference data standard's macros, programs and templates folders to their respective folders in the custom data standard's folder structure.

Next, use the SAS Clinical Standards Toolkit macros to create empty template data sets for all 13 of the required data sets. This will create zero-observation data sets, with all of the correct metadata, ready to be populated with the required data. The following code will create a template standards data set.

```
%cst_setStandardProperties(_cstStandard=CST-FRAMEWORK,_cstSubType=initialize);

%cstutil_setcstgroot;

libname newmeta "&_cstgroot./standards/study-nicardipine-sah-01-1.0/control";

%cst_createDSfromtemplate( _cstStandard=CST-FRAMEWORK,_cstType=standards,
_cstSubType=registeredstandards,_cstOutputDS=newmeta.standards );
```

Table 1 shows the values for the cst_CreateDSFromTemplate macro for each of the 13 data sets. Keep in mind that the appropriate libref should be specified for each of the data sets in the _cstOutputDS parameter.

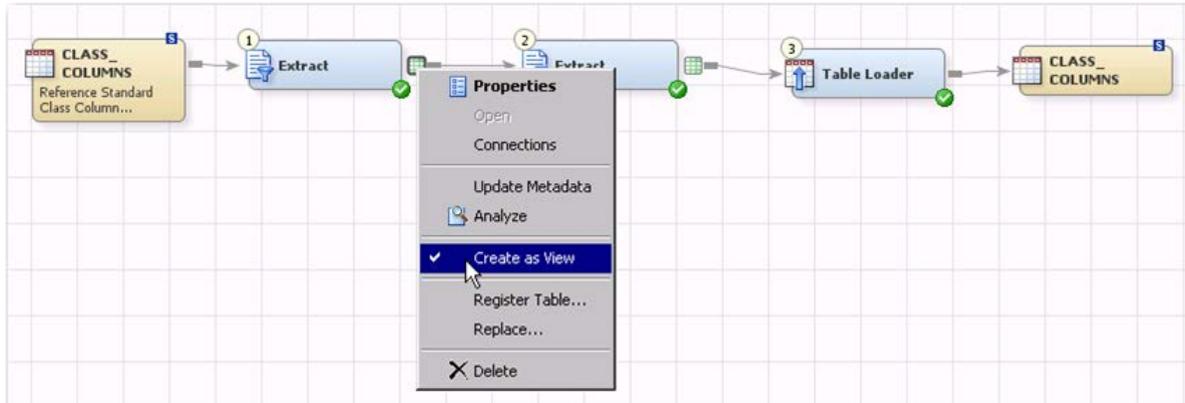
Table 1: Parameters for cst_CreateDSFromTemplate Macro

Data Set (_cstOutputDS)	_cstStandard	_cstType	_cstSubType	Folder
standards	CST-FRAMEWORK	standards	registeredstandards	control
standardsasreferences	CST-FRAMEWORK	standards	registeredsasreferences	control
standardlookup	CST-FRAMEWORK	cstmetadata	lookup	control
standardmacrovariables	CST-FRAMEWORK	cstmetadata	macrovariables	control
standardmacrovariabledetails	CST-FRAMEWORK	cstmetadata	macrovariabledetails	control
messages	CST-FRAMEWORK	messages		messages
class_columns	CDISC-SDTM	classmetadata	column	metadata
class_tables	CDISC-SDTM	classmetadata	table	metadata
reference_columns	CDISC-SDTM	referencemetadata	column	metadata
reference_tables	CDISC-SDTM	referencemetadata	table	metadata
validation_master	CST-FRAMEWORK	control	validation	validation/control
validation_stdref	CST-FRAMEWORK	referencecontrol	standardref	validation/control
validation_domainsbycheck	CDISC-SDTM	referencecontrol	checktable	validation/control

POPULATE DATA SETS

Populating the data sets required for the custom data standard can be done using Base SAS programming or it can be done inside SAS Clinical Data Integration. When using Base SAS programming, SAS recommends making the libraries that point to the reference data standard read-only so that no one can accidentally overwrite the reference data standard's data sets. When creating jobs in SAS Clinical Data Integration to map values from the reference data standard, the user will be using tables with the same name (but different libraries and physical locations) for both their source and target. In order to avoid errors, be certain that the work tables created during the job are not views, but instead are actual work library data sets. To do this, right click on each work table node on transformations in your job and deselect "Create as View" (there should be no check mark next to this option). See Figure 3 for an illustration of this table option.

FIGURE 3: DESELECTING CREATE AS VIEW FROM WORK TABLES



PhUSE 2014

In order to understand how to populate the data sets and files, it is important to understand the purpose of each data set and file. Table 2 describes the 13 data sets that define a data standard, and has been adapted from the data set descriptions found in the SAS Clinical Standards Toolkit 1.6 User's Guide. I recommend reviewing these data sets for an existing CDISC data standard in the CST Global Library to help get an understanding of the contents and structure of each of these data sets.

TABLE 2: DESCRIPTION OF DATA SETS THAT DEFINE A DATA STANDARD

Folder	Data Set	Description
control	standards	The standards data set is a single-record file that provides metadata about the standard. It is appended to the master standards data set and used by the SAS Clinical Standards Toolkit framework to store information about the standard version when the standard is registered.
	standardsasreferences	The standardsasreferences data set specifies a set of library and file records that are used by most processes that are provided with the SAS Clinical Standards Toolkit implementation of the standard. It specifies the name, location, and metadata for all 13 of the data sets described in this table, the two properties files, the templates, and the macro autocall path.
	standardlookup	The standardlookup data set provides a mechanism to capture valid values for discrete variables in the SAS Clinical Standards Toolkit metadata files. This data set supports such tasks as validating the content of the SAS Clinical Standards Toolkit metadata files and providing selectable values in the user interfaces of other tools and solutions. It almost serves like controlled terminology, specifying possible values for several columns within the data sets described in this table, as well as specifying which value is the default. One example is Type column used in class_columns and reference_columns: the possible values are "C" or "N", and the default is "C".
	standardmacrovariables	The standardmacrovariables data set contains metadata about global macro variables used in the data standard. This includes whether the macro variable is required, what type of value it has (such as INTEGER, DATASET), and where its value comes from (initialize or validation properties file).
	standardmacrovariabledetails	The standardmacrovariabledetails data set contains possible values for the global macro variables used in the data standard, as well as indicators of their default values.
messages	messages	The messages data set provides error messaging for all of the validation checks in the validation_master data set.
metadata	class_columns	The class_columns data set identifies the full set of column definitions used in the data standard's domains. This is a generalized definition of column metadata that is not domain-specific, such as __TERM, which may be used in several domains (AETERM, MHTERM, etc.). Included in the column metadata is the class the column belongs to, such as EVENTS, FINDINGS, INTERVENTIONS, IDENTIFIERS, and TIMING.
	class_tables	The class_tables data set identifies the full set of column classes used by the data standard's domain columns. Some examples are EVENTS, FINDINGS, INTERVENTIONS, IDENTIFIERS, and TIMING.
	reference_columns	Part of the definition of each standard is the itemization of the columns in each domain that defines the SAS representation of that standard and version. The reference_columns data set captures column-level metadata for the specific columns in the domains that are supported for the data standard. This information is different for each version of the data standard.
	reference_tables	Part of the definition of each standard is the itemization of the domains that define the SAS representation of that standard and version. The reference_tables data set captures table-level metadata for the specific domains that are supported for the data standard. This information is different for each version of the data standard.
validation/control	validation_master	Each standard that supports validation has a validation_master data set that provides the full set of validation checks defined for that standard. These checks validate the domain structure and content for each version.
	validation_stdref	The validation_stdref data set contains additional information about each of the checks in the validation_master data set. This data set provides additional information about the origin of the check and any supporting documentation about the check.
	validation_domainsbycheck	The validation_domainsbycheck contains records for each domain that is to be validated by each check in the validation_master data set. This provides the ability to subset checks that are applicable to specific domains.

PhUSE 2014

The first data set to get familiar with is the standards data set. This describes the highest level metadata for the user's data standard, and it is important to understand what each variable represents. Table 3 describes each of the variables in this data set. This table has been adapted from Table 3.1 from the SAS Clinical Standards Toolkit 1.6 User's Guide.

TABLE 3: VARIABLES IN THE STANDARDS DATA SET

Column Name	Column Length	Description
standard	(\$20)	The name of the registered standard. Think of this as the standard type or family. For CDI, only the following values are valid: CDISC-SDTM, CDISC-ADAM, CDISC-SEND.
mnemonic	(\$4)	A short mnemonic for the standard, usually SDTM, ADAM, or SEND.
standardversion	(\$20)	The version number or name of the registered standard. Must be unique within the standard. This is the value that should clearly identify the user's custom data standard.
groupname	(\$20)	The standard group across versions, such as STDM or TERMINOLOGY. For custom data standards, this should be the data standard type that the custom standard is based on.
groupversion	(\$20)	The version of the groupname, often the same as standardversion. For custom data standards, this should be the specific version of the data standard that the custom standard is based on.
comment	(\$200)	A description of the registered standard version.
rootpath	(\$200)	The root path for the standard version's directory in the global standards library.
studylibraryrootpath	(\$200)	The root path to the study repository. This can be used to initialize the studyRootPath and studyOutputPath global macro variables and to use relative paths to study library subfolders. By default, this is set to the sample library that is associated with each standard provided with the SAS Clinical Standards Toolkit.
controlsubfolder	(\$200)	The control folder path (relative to rootpath). This value provides the location of data sets that are required for standard registration (such as Standards and StandardSASReferences).
templatesubfolder	(\$200)	The template folder path (relative to rootpath). This value provides the location of data sets that are specific to the standard that serve as templates for standard-specific processes.
isstandarddefault	(\$1)	A value that identifies whether the version is the default for the standard. More than one version can be registered and there can still be a default version. Valid values are Y, N.
iscstframework	(\$1)	A value that identifies whether the standard version is part of the framework. This column can be used to subset the list of registered standards. Valid values are Y and N.
isdatastandard	(\$1)	A value that identifies whether the standard version is a data standard. For example, CDISC SDTM versions are data standards, and CDISC Controlled Terminology is not. Valid values are Y and N.
supportvalidation	(\$1)	A value that identifies whether the standard version supports validation. Valid values are Y and N.
isxmlstandard	(\$1)	A value that identifies whether the standard version is based on XML. CDISC SDTM is not, and CDISC CRT-DDS is based on XML. Valid values are Y and N.
importxsl	(\$200)	If the standard version is based on XML, then this is the path to the XSL file to import the XML into the SAS representation.
exportxsl	(\$200)	If the standard version is based on XML, then this is the path to the XSL file to export the XML file.
schema	(\$200)	If the standard version is based on XML, then this is the path to the XML schema document that can be used to validate the XML.
productrevision	(\$10)	The revision of the standard and standardversion that is currently installed. For standards provided with SAS Clinical Standards Toolkit, this is the version of SAS CST. For custom data standards, this can be a supplemental version identifier.

Table 4 below provides the values of the standards data set for the CDISC SDTM 3.1.3 data standard alongside the values used for the NICSAH-01 data standard. Notice that rootpath is missing, but when the data standard is registered to SAS Clinical Standards Toolkit, this value will be automatically populated in the master standards data set.

TABLE 4: VALUES FOR STANDARDS DATA SET

Column Name	Column Length	Values for CDISC-SDTM 3.1.3 Standard	Values for Custom Data Standard (Study NICSAH-01)
standard	(\$20)	CDISC-SDTM	CDISC-SDTM
mnemonic	(\$4)	SDTM	SDTM
standardversion	(\$20)	3.1.3	STUDY-NICSAH-01
groupname	(\$20)	SDTM	SDTM
groupversion	(\$20)	3.1.3	3.1.3

PhUSE 2014

Column Name	Column Length	Values for CDISC-SDTM 3.1.3 Standard	Values for Custom Data Standard (Study NICSAH-01)
comment	(\$200)	CDISC SDTM V3.1.3	Study-specific data standard for study NICSAH-01
rootpath	(\$200)		
studylibraryrootpath	(\$200)	&_cstSRoot./cdisc-sdtm-3.1.3-1.6/sascstdemodata	&_cstSRoot./cdisc-sdtm-3.1.3-1.6/sascstdemodata
controlsubfolder	(\$200)	control	control
templatesubfolder	(\$200)	templates	templates
isstandarddefault	(\$1)	N	N
iscstframework	(\$1)	N	N
isdatastandard	(\$1)	Y	Y
supportvalidation	(\$1)	Y	Y
isxmlstandard	(\$1)	N	N
importxsl	(\$200)		
exportxsl	(\$200)		
schema	(\$200)		
productrevision	(\$10)	1.6	1.0

The reference_tables data set describes each of the domains the user intends to define for a custom data standard. There should be one record per domain. It would be appropriate to define each of the individual SUPPQUAL domains here. Table 5 describes all of the variables in the reference_tables data set; reviewing this data set for an existing CDISC SDTM data standard in the CST Global Library alongside this table can be very helpful. This table has been adapted from Table 7.1 in the SAS Clinical Standards Toolkit 1.6 User's Guide.

TABLE 5: VARIABLES IN THE REFERENCE_TABLES DATA SET

Column	Column Length	Description
sasref	\$8	The SAS libref that refers to the table in the SAS Clinical Standards Toolkit process. This value should match the value of the SASReferences.sasref field, where type=referencemetadata and subtype=table. This column is required.
table	\$32	The name of the tabulation domain or analysis data set being defined in the standard. The value must conform to SAS naming conventions (If the label text contains single quotation marks, use double quotation marks around the label, or use two single quotation marks in the label text and surround the string with single quotation marks). This column is required.
label	\$200	The label of the domain being defined in the standard. The value must conform to SAS data set label naming conventions. This column is required for standards from which define.xml metadata is derived.
class	\$40	The observation class in the standard. Example CDISC SDTM values are Events, Findings, Interventions, Relates, Special Purpose, and Trial Design. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
xmlpath	\$200	The path to the SAS transport file. This path can be specified as a relative path. The value can be used when creating define.xml to populate the value for the def:leaf xlink:href link to the domain file. The value should be the pathname and filename of the SAS transport file relative to the location of define.xml file. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
xmltitle	\$200	The title of the SAS transport file. The value can be used when creating a define.xml file to populate the value for the def:leaf def:title value. It can provide a meaningful description, label, or location of the domain leaf (for example, crt/datasets/Protocol 1234/AE.xpt). This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
structure	\$200	The description of the general structure of the table. An example value is one record per event per subject. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
purpose	\$20	The description of the general purpose of the table. Examples are Tabulation (required for CDISC SDTM) and Analysis (required for CDISC ADaM). This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.

PhUSE 2014

Column	Column Length	Description
keys	\$200	A space-delimited string of keys that captures the table columns that uniquely define records in the table. This set of keys can also define the sort order of records in the table. Example is STUDYID USUBJID. This column is expected to support SAS Clinical Standards Toolkit functionality. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
state	\$20	A description of the table state, such as Draft or Final. This column is optional.
date	\$20	A meaningful, distinguishing date that describes the table, such as the release date, the creation date, or the modified date. This column is optional.
standard	\$20	The name of the registered standard. Think of this as the standard type or family; for CDI, only the following values are valid: CDISC-SDTM, CDISC-ADAM, CDISC-SEND
standardversion	\$20	The version number or name of the registered standard. Must be unique within the standard. This is the value that should clearly identify the custom data standard.
standardref	\$200	Any reference to an associated standard definition, implementation guide, schema, and so on that provides additional information about the table or describes the table in greater detail. This column is optional.
comment	\$500	Any character string that provides comments relevant to the table. This column is optional. (This comment will not be carried through to the define.xml file; it is purely informational.)

The reference_columns data set further describes each of the user's domains by providing metadata about each column in each domain for the custom data standard. Table 6 describes all of the variables in the reference_columns data set. Reviewing this data set for an existing CDISC SDTM data standard in the CST Global Library alongside this table can be very helpful. This table has been adapted from Table 7.2 in the SAS Clinical Standards Toolkit 1.6 User's Guide.

TABLE 6: VARIABLES IN THE REFERENCE_COLUMNS DATA SET

Column	Column Length	Description
sasref	\$8	The SAS libref that refers to the table containing the column in the SAS Clinical Standards Toolkit process. This value should match the value of the SASReferences.sasref field, where type=referencemetadata and subtype=column. This column is required.
table	\$32	The name of the tabulation domain or analysis data set being defined in the standard. The value must conform to SAS naming conventions. This column is required.
column	\$32	The name of the column in the table. The value must conform to SAS naming conventions. This column is required.
label	\$200	The label of the column. The value must conform to SAS variable label naming conventions. This column is required for standards from which define.xml metadata is derived.
order	8.	The order of the columns in each table. Values must be integers >0 and unique in each table. This column is required.
type	\$1	The SAS type, N for numeric, C for character. This column is required.
length	8.	The length of the column. Numeric columns have a length of 8. This column is required.
displayformat	\$32	The display format for numeric variables. For example, 8.2 indicates that floating-point variable values should be displayed to the second decimal place. This value is optional and not relevant for all standards.
xmldatatype	\$8	The data type of the column, as it is defined in the define.xml file. Values are integer float date datetime time text. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
xmlcodelist	\$32	A SAS format name that is used to assess conformance to controlled terminology. This value does not have a \$ prefix for character formats and does not have the trailing period. This value is also the codelist name in the define.xml file. The SAS format name must be in the format search path for successful column-value validation. This column is optional.
core	\$10	The value indicates whether the column is required. Sample CDISC SDTM values are Req (required), Exp (expected), Perm (permissible), and Dep (deprecated). This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.

PhUSE 2014

Column	Column Length	Description
origin	\$40	Information about the source of the column. Values can include CRF page numbers and derived or variable references. Values are user extensible. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
role	\$200	Space-delimited column classification. Examples are Identifier, Topic, Qualifier, Timing, Selection, and Analysis. Columns can have multiple roles. This column is not strictly required by SAS Clinical Standards Toolkit, but should be populated for ADAM- and SDTM-based data standards.
term	\$80	The value contains information about the terminology associated with the column, such as the specific controlled term(s) or codelist name that is used to assess conformance to controlled terminology. This column is optional.
algorithm	\$1000	Imputation or computation method to derive the column value. This column is optional.
qualifiers	\$200	Space-delimited string containing supplemental column attributes. Example CDISC SDTM values are MIXEDCASE, UPPERCASE, DATETIME, and DURATION. This column is optional.
standard	\$20	The name of the registered standard. I think of this as the standard type or family; for CDI, only the following values are valid: CDISC-SDTM, CDISC-ADAM, CDISC-SEND
standardversion	\$20	The version number or name of the registered standard. Must be unique within the standard. This is the value that should clearly identify your custom data standard.
standardref	\$200	Any reference to an associated standard definition, implementation guide, schema, and so on that provides additional information about the column or describes the column in greater detail. This column is optional.
comment	\$1000	Any character string that provides comments relevant to the column. This column is optional. (This comment will not be carried through to the define.xml file; it is purely informational.)

As mentioned before, many of the data sets' contents may be copied from the reference data set with little or no modification. Some of these data sets contain the standard and standardversion variables; these values should be modified to match the user's custom data standard, but the remaining variable values from the reference data standard may be copied as-is. When using Base SAS for this step, the following code is an example that will preserve the metadata of the template data set already created, while inserting the values from the reference data standard.

```
libname mastmeta 'C:\cstGlobalLibrary\standards\cdisc-sdtm-3.1.3-1.6\metadata'
    access=readonly;
libname custmeta 'C:\cstGlobalLibrary\standards\study-nicardipine-sah-01-
    1.0\metadata';

proc sql;
    insert into custmeta.class_tables select * from mastmeta.class_tables;
    update custmeta.class_tables set standardversion="STUDY-NICSAH-01";
quit;
```

TABLE 7: LIST OF DATA SETS TO COPIED FROM REFERENCE DATA STANDARD

Copy with No Modification	Copy and Modify standard and standardversion
standardmacrovariables messages validation_domainsbycheck validation_master* validation_stdref*	standardlookup standardsasreferences class_columns class_tables

**Although these data sets do have the standard and standardversion variables, they indicate the version of the data standard at which the specific compliance check originated, and do not indicate the version of the current data standard.*

The standardmacrovariabledetails data set is a slightly different case, because its structure is one record per macro variable. It does contain the values of standard and standardversion, but they are distinct records. The variable values for this data set should be copied from the reference data standard, but the observations where macrovariable=_cstStandard and _cstStandardVersion should have the value of macrovalue changed to match the standard and standardversion for your custom data standard. See Figure 4 for a snapshot of these records from the CDISC SDTM 3.1.3 data standard.

FIGURE 4: EXAMPLE OF STANDARDMACROVARIABLEDETAILS DATA SET

	macrovariable	macrovalue	macrovaluelabel	default
33	_cstMetricsTimer	0	Off	N
34	_cstMetricsTimer	1	On	Y
35	_cstStandard	CDISC-SDTM		Y
36	_cstStandardVersion	3.1.3		Y
37	_cstSubjectColumns	studyid usubjid		Y
38	_cstTableMetadata	work._csttablemetadata		Y

MODIFY PROPERTIES FILES

There are two properties files in the programs folder, named initialize.properties and validation.properties. Both of these properties files set macro variable default values that are used by SAS Clinical Standards Toolkit processes. These macro variables are all described in the standardmacrovariables data set, and the possible values and selected default values for the data standard are detailed in the standardmacrovariabledetails data set. The initialize.properties file contains five macro variables and their default values (versions 1.5 and 1.6 of SAS Clinical Standards Toolkit), but this file must be modified using a text editor to have the correct value of standard and standardversion. Figure 5 shows the initialize.properties file for the CDISC SDTM 3.1.3 data standard and Figure 6 shows the file for the NISCAH-01 study-specific data standard.

FIGURE 5: INITIALIZE.PROPERTIES FILE FOR CDISC-SDTM 3.1.3 DATA STANDARD

```
initialize.properties - Notepad
File Edit Format View Help
_cstStandard=CDISC-SDTM
_cstStandardVersion=3.1.3
_cstSubjectColumns=studyid usubjid
_cstTableMetadata=work._csttablemetadata
_cstColumnMetadata=work._cstcolumnmetadata
```

FIGURE 6: INITIALIZE.PROPERTIES FILE FOR NISCAH-01 STUDY DATA STANDARD

```
initialize.properties - Notepad
File Edit Format View Help
_cstStandard=CDISC-SDTM
_cstStandardVersion=STUDY-NISCAH-01
_cstSubjectColumns=studyid usubjid
_cstTableMetadata=work._csttablemetadata
_cstColumnMetadata=work._cstcolumnmetadata
```

The validation.properties file contains the remaining macro variables and their default values. For a typical study-specific data standard, these values would not be modified. However, it is possible to review them and modify values as needed. When changing any of these values, the standardmacrovariabledetails data set should also be updated to change the default “Y” and “N” flags on the modified macro variables. Figure 7 shows the validation.properties file for the CDISC SDTM 3.1.3 data standard.

FIGURE 7: VALIDATION.PROPERTIES FILE FOR THE CDISC SDTM 3.1.3 DATA STANDARD

```
validation.properties - Notepad
File Edit Format View Help
_cstCheckSortOrder=_DATA_
_cstMetrics=1
_cstMetricsDS=work._cstmtrics
_cstMetricsNumSubj=1
_cstMetricsNumRecs=1
_cstMetricsNumChecks=1
_cstMetricsNumBadChecks=1
_cstMetricsNumErrors=1
_cstMetricsNumWarnings=1
_cstMetricsNumNotes=1
_cstMetricsNumStructural=1
_cstMetricsNumContent=1
_cstMetricsCntNumSubj=0
_cstMetricsCntNumRecs=0
_cstMetricsCntNumChecks=0
_cstMetricsCntNumBadChecks=0
_cstMetricsCntNumErrors=0
_cstMetricsCntNumWarnings=0
_cstMetricsCntNumNotes=0
_cstMetricsCntNumStructural=0
_cstMetricsCntNumContent=0
_cstMetricsTimer=1
```

QUALITY CHECKS ON DATA SETS

Now all of the data sets have been populated, the properties files have been reviewed and updated, and the data standard is ready. However, before moving forward to register the standard with SAS Clinical Standards Toolkit, now is a good time to stop and do a few quality checks of the 13 data sets. This process can be customized to check the items the user finds most important. The following is a list of checks SAS recommends, at a minimum.

1. PROC CONTENTS of all 13 data sets
 - Verify no data sets have zero observations.
 - As you get more familiar with these data sets, you will recognize an expected number of observations and variables for each data set, so you can identify potential problems easily.
2. Verify that the values for standard and standardversion are identical in all data sets that contain these values.
 - These values must be identical in all data sets that contain them (don't forget to check the _cstStandard and _cstStandardVersion records in the standardmacrovariabledetails data set).
3. Verify that the standard/standardversion combination used for your data standard is not already in the master standards data set.
 - Although this was something you already should have checked for, now is a good time to check one more time.

4. Check that required variables are non-missing for all observations.
 - For a typical study-specific data standard, SAS recommends checking the reference_tables and reference_columns data sets for missing required values. See Table 8 for required values. The other data sets will have been mostly copied from the reference data standard and should not have missing required data.

TABLE 8: REQUIRED VARIABLES IN REFERENCE_TABLES AND REFERENCE_COLUMNS DATA SETS

reference_tables		reference_columns	
sasref	structure*	sasref	xmldatatype*
table	purpose*	table	core*
label*	keys*	column	origin*
class*	standard	label*	role*
xmlpath*	standardversion	order	standard
xmltitle*		type	standardversion
		length	

**These variables are not required to be non-missing by SAS Clinical Standards Toolkit, but for the purpose of study-specific custom data standards, SAS would recommend that these variables also be non-missing.*

5. Verify that the set of domains described in both reference_tables and reference_column is identical
 - Get a list of the domains described in each data set and compare them. If there are domains defined in only one of the data sets, the domain definition is incomplete.
 - There are two options for dealing with an incomplete domain definition: remove it from the one data set it does appear in, or else add metadata for the domain to the data set in which its definition is missing.
 - When registering the standard to SAS Clinical Standards Toolkit, every domain should have both its table and column metadata defined.

REGISTER CUSTOM DATA STANDARD TO SAS CLINICAL STANDARDS TOOLKIT

Finally, the custom data standard is ready to be registered to SAS Clinical Standards Toolkit. The code to register a data standard is simple. It may be run from the Code Editor in SAS Clinical Data Integration, a job in SAS Clinical Data Integration using a user-written or custom generated transformation, or from a Base SAS session that can access the user's SAS Clinical Standards Toolkit installation. The following code should be used to register a custom data standard, with <custom data standard folder name> replaced with the exact folder name for the custom data standard. For the NICSAH-01 study data standard, this would be study-nicardipine-sah-01-v1.0.

```
%cst_setStandardProperties(_cstStandard=CST-FRAMEWORK,_cstSubType=initialize);

%cstutil_setcstgroot;

%cst_registerStandard(_cstRootPath=
%nrstr(&_cstGRoot./standards/<custom data standard folder name>),
_cstControlSubPath=control,_cstStdDSName=standards,
_cstStdSASRefsDSName=standardsasreferences,_cstStdLookupDSName=standardlookup);
```

This code will append the observations from the standards, standardsasreferences, and standardlookup data sets in the custom data standard's control folder to the data sets with the same name in the Global Standards Library's metadata folder. These data sets are referred to as the master metadata data sets because they contain the metadata for each data standard registered to SAS Clinical Standards Toolkit.

After running the code, check the SAS log for any issues. If the registration was successful, a message should appear in the log that states that the data standard was registered. If this message appears, SAS recommends opening each of the data sets in the master metadata folder to verify that all of them contain the records for the custom data standard. If any one of them does not have records for the custom data standard, the registration was not complete.

There may be errors or other issues in the SAS log after running the code to register the standard. In this case, the registration of the custom data standard was likely unsuccessful and/or incomplete. In this case SAS also recommends opening each of the data sets in the master metadata folder to see if any of them contain records for the custom data standard.

In either of these cases of incomplete or unsuccessful registration of a data standard where at least some records have been written to the master metadata data sets, the data standard must be unregistered from SAS Clinical Standards Toolkit, then the issues with the data standard should be fixed, and then the data standard can be registered again.

In order to unregister a data standard, use the following SAS code, with the value of standard and standardversion modified for the custom data standard. For the NICSAH-01 study data standard, these would be _cstStandard=CDISC-SDTM, _cstStandardVersion=STUDY-NICSAH-01-v1.0.

PhUSE 2014

```
cst_setStandardProperties(_cstStandard=CST-FRAMEWORK,_cstSubType=initialize);  
  
%cstutil_setcstgroot;  
  
%cst_unregisterStandard(_cstStandard=<standard type>, _cstStandardVersion=<version>);
```

IMPORT CUSTOM DATA STANDARD INTO SAS CLINICAL DATA INTEGRATION

The custom data standard has been registered to SAS Clinical Standards Toolkit, and now it is ready to be imported into SAS Clinical Data Integration. This process is the same as registering any CDISC data standard in SAS Clinical Data Integration. On the Clinical Administration tab, select the Data Standards folder, right click, and select Import. Select the data standard type for the custom data standard and select Next. On the following screen, the custom data standard version will appear in the list of available versions. Continue the process to finish the import.

Next, take a look at the tables under Column Groups, Domain Templates, and Validation Datasets. Be sure that the five Column Groups are present (identifiers, findings, events, interventions, timing) and that it is possible to open and view the properties. Also verify that all of the defined domains in the custom data standard appear under Domain Templates. Open one or two of them and verify that the Column Properties look correct. Finally, verify that the merged_validation table exists under Validation Datasets, and open it to verify that it contains compliance check information.

Any problems with the imported data standard shows that the underlying data sets must have some problem. Delete the data standard from SAS Clinical Data Integration, and investigate the data sets, including the master metadata data sets. Once these issues have been found and fixed, unregister the data standard from SAS Clinical Standards Toolkit, then re-register it, and continue with importing the data standard into SAS Clinical Data Integration again.

CONCLUSION

The CDISC data standards have been adopted by the pharmaceutical industry, but there remain relics of legacy data standards within organizations, and no clinical study's data adheres perfectly to the CDISC data standards. SAS Clinical Data Integration provides metadata for several CDISC data standards. These provided data standards may be used as a template or reference for creating custom data standards at the organizational, therapeutic area, and study level, maximizing and optimizing use of SAS Clinical Data Integration.

REFERENCES

- SAS Institute, Inc. 2014. *SAS Clinical Data Integration 2.5 User's Guide*. Cary, NC: SAS Institute, Inc.
<http://support.sas.com/documentation/cdl/en/clindiug/66867/PDF/default/clindiug.pdf>
- SAS Institute, Inc. 2014. *SAS Clinical Standards Toolkit 1.6 User's Guide*. Cary, NC: SAS Institute, Inc.
<http://support.sas.com/documentation/cdl/en/clinstdtktug/66870/PDF/default/clinstdtktug.pdf>
- SAS Institute, Inc. 2014. *SAS Clinical Standards Toolkit 1.6 Macro API Documentation*. Cary, NC: SAS Institute, Inc.
<http://support.sas.com/documentation/onlinedoc/clinical/1.6/cst1.6-macro-api/index.html>

ACKNOWLEDGMENTS

Thank you to Julie Maddox, Michael Kihullen, and Lex Jansen for always being so helpful and patient with my many CDI and CST questions. Thank you also to Donna Dutton, Sharon Trevoy, Gene Lightfoot, Chris Butler, Kevin Alderton, Bill Gibson, Adam LaManna, and Danny Martinez for the valuable feedback. Finally, thank you to Angela Lightfoot for continued support and encouragement.

CONTACT INFORMATION

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

Melissa R. Martinez
SAS Institute, Inc.
720 SAS Campus Drive
Cary, North Carolina 27513, USA
Work Phone: +1 (919) 531-9277
Email: Melissa.Martinez@sas.com

Brand and product names are trademarks of their respective companies.