SAS® Clinical Standards Toolkit 1.5

Getting Started

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *SAS® Clinical Standards Toolkit 1.5: Getting Started*. Cary, NC: SAS Institute Inc.

SAS® Clinical Standards Toolkit 1.5: Getting Started

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

May 2013

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit **support.sas.com/bookstore** or call 1-800-727-3228.

Contents

Using This Book

Audience

The SAS Clinical Standards Toolkit and this book are designed for SAS programmers in the clinical research industry. Programmers must have a good understanding of the SAS macro language and must also be knowledgeable of evolving industry data standards, such as the Clinical Data Interchange Standards Consortium (CDISC).

Requirements

This book does not attempt to provide instruction to install the SAS Clinical Standards Toolkit. It is assumed that the SAS Clinical Standards Toolkit has been successfully installed by a SAS Administrator and that the installation has been verified to have been installed correctly and to be properly functioning.

Recommended Reading

Here is the recommended reading list:

- *SAS Clinical Standards Toolkit: User's Guide*

  This book provides more comprehensive information about the SAS Clinical Standards Toolkit.

- *SAS Clinical Standards Toolkit: Installation Qualification*

For a complete list of SAS books, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Book Sales Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

1

Introduction to the SAS Clinical Standards Toolkit

Key SAS Clinical Standards Toolkit Functionality

This book provides a brief introduction to the SAS Clinical Standards Toolkit. The SAS Clinical Standards Toolkit serves two primary purposes:

- Provide SAS implementations, based in SAS, of evolving clinical data standards.
- Provide a framework that exploits these standards to meet common clinical research analysis and submission requirements.

Through the SAS Clinical Standards Toolkit 1.5 release, the SAS Clinical Standards Toolkit supports only standards developed by the Clinical Data Interchange Standards Consortium (CDISC). However, there is nothing in the product design that limits support of other standards, and it is anticipated that both current users and future SAS Clinical Standards Toolkit releases will support non-CDISC clinical-related standards.

The SAS Clinical Standards Toolkit is an open-source solution. Most of the code is either Base SAS or SAS macro code. For XML-based standards, some JAVA and XSLT code is used. The SAS Clinical Standards Toolkit is a SAS solution for SAS users.

Given that study design, data collection, and analysis and submission requirements of each research protocol are unique, and that the SAS Clinical Standards Toolkit provides libraries of open-source SAS macros and code, user customization is expected and encouraged.

The SAS Clinical Standards Toolkit File Roadmap

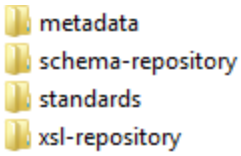
The SAS Clinical Standards Toolkit files can be aggregated into three primary groups:

- the SAS Clinical Standards Toolkit global standards library
- SAS program files (!sasroot)
- the SAS Clinical Standards Toolkit sample library

The primary function of the SAS Clinical Standards Toolkit global standards library is to provide the metadata that defines each supported SAS Clinical Standards Toolkit standard. This location is specified during product installation and can be subsequently moved to another location. By default, on Microsoft Windows, this location is set to the `c:\cstGlobalLibrary` directory.

Here are the top-level subfolders:

Figure 1.1 *Global Standards Library Top Folder Hierarchy*

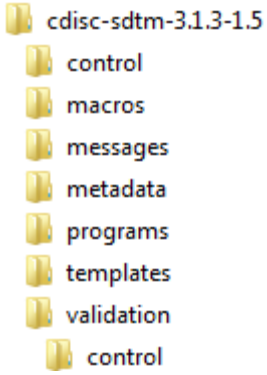


The metadata folder contains key metadata about all supported standards and can be updated with each new product release. The metadata folder can also be updated each time you elect to register a new standard or version of a standard. The schema-

repository and xsl-repository folders support XML-based standards (such as CDISC-ODM and CDISC CRT-DDS). The standards folder contains a subfolder hierarchy for each supported standard.

This folder hierarchy illustrates the types of metadata supporting the CDISC SDTM 3.1.3 standard:

Figure 1.2 *Global Standards Library Folder Hierarchy, CDISC SDTM 3.1.3*



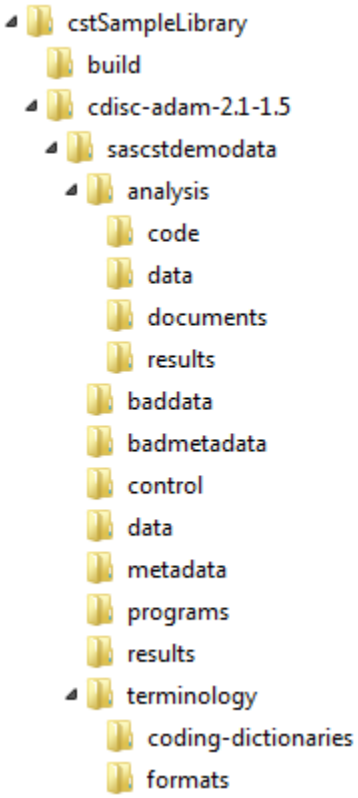
The **control** folder contains data sets that provide general metadata about the standard. The **macros** folder contains SAS macros for standards that are included in the SAS autocall path when working with that standard. The **messages** folder contains messages used in reporting or process results for a specific standard. The **metadata** folder contains the "gold-standard" definition of the standard as interpreted and implemented by the person who registered the standard. The **programs** folder contains programs and property files supporting general use of the standard. The **validation** folder and the **control** folder contain validation check metadata for those standards supporting validation of standard domains or data sets.

Installation of the SAS Clinical Standards Toolkit places files in the !sasroot tree for the common SAS Clinical Standards Toolkit framework files. The generic (cross-standard or standard-independent) framework SAS macros are installed to the **`!sasroot/cstframework/sasmacros`** directory (Microsoft Windows). The SAS config file has been altered when the SAS Clinical Standards Toolkit was installed to include this folder, by default, in the SASautos path, meaning that the framework macros are available to you whenever and however you start SAS referencing the default config file.

In the SAS Clinical Standards Toolkit sample library, there is a sample folder for each standard. A sample folder contains the files that represent a sample SAS implementation of the specific standard, sample study data sets, and sample programs and results that illustrate the SAS Clinical Standards Toolkit functionality.

Here is the CDISC ADaM 2.1 sample folder hierarchy:

Figure 1.3 *Folder Hierarchy for CDISC ADaM 2.1 Sample*



The folder hierarchy might be different for other standards. For example, the CDISC ADaM **`analysis`** folder and subfolders are unique to ADaM. The **`cstSampleLibrary`** folder merely provides a location for the sample SAS implementation. Your files can be located anywhere accessible by SAS. You do not need to conform to the folder-naming convention illustrated in the CDISC ADaM sample.

Key SAS Clinical Standards Toolkit Components

Framework macros and SAS macros specific to a standard, as described above, provide the foundation for the SAS Clinical Standards Toolkit functionality. Three other SAS Clinical Standards Toolkit components are also noteworthy:

- driver modules
- properties
- SASReferences data set

The SAS Clinical Standards Toolkit provides a set of driver modules (non-macro SAS code) that illustrate all of the basic Toolkit functions using sample data provided by SAS. These drivers can serve as templates for the Toolkit processes you want to develop. These drivers might be submitted as batch or interactive processes. They can be found in the programs subfolder for each standard (for the relative location of this folder, see Figure 1.3 on page 4). For a more complete description of what these drivers do, see “Using the SAS Clinical Standards Toolkit” on page 7.

The SAS Clinical Standards Toolkit also provides a series of framework properties files and properties files specific to a standard (such as initialization, validation, and report properties files). You can use these files to both define global macro variables and to set their default values. These typically can be found in the same folders as the driver modules. Use of these property files is optional but recommended, and you can add your own global macro variables as needed.

Perhaps the single most important file used by the SAS Clinical Standards Toolkit is the SASReferences data set. The SASReferences data set captures all the input and output file library and file references associated with any given process. Here is a sample:

Figure 1.4 *Sample SASReferences Data Set (CDISC SDTM 3.1.2 Validation Process)*

standard	standardversion	type	subtype	SASref	reftype	iotype	filetype	allowoverwrite	path	order	memname
CDISC-SDTM	3.1.2	autocall		autocall	fileref	input	folder	N		1	
CDISC-SDTM	3.1.2	control	reference	srccntl	libref	input	dataset	N	&studyRootPath/control	.	sasreferences.sas7bdat
CDISC-SDTM	3.1.2	control	validation	srccntl	libref	input	dataset	N	&studyRootPath/control	.	validation_control.sas7bdat
CDISC-SDTM	3.1.2	fmtsearch		fmts	libref	input	catalog	N	&studyRootPath/terminology/formats	1	formats.sas7bcat
CDISC-SDTM	3.1.2	lookup		lookup	libref	input	dataset	N		.	
CDISC-SDTM	3.1.2	messages		messages	libref	input	dataset	N	&_cstGRoot/standards/cdisc-sdtm-3.1.2-	2	messages.sas7bdat
CDISC-SDTM	3.1.2	properties	initialize	initprop	fileref	input	file	N	&_cstGRoot/standards/cdisc-sdtm-3.1.2-	1	initialize.properties
CDISC-SDTM	3.1.2	properties	validation	valprop	fileref	input	file	N	&studyRootPath/programs	2	validation.properties
CDISC-SDTM	3.1.2	referencecontrol	checktable	refcntl	libref	input	dataset	N		.	
CDISC-SDTM	3.1.2	referencecontrol	standardref	refcntl	libref	input	dataset	N		.	
CDISC-SDTM	3.1.2	referencecontrol	validation	refcntl	libref	input	dataset	N		.	
CDISC-SDTM	3.1.2	referencecterm		ctref	libref	input	dataset	N	&studyRootPath/terminology/coding-dicti	1	meddra.sas7bdat
CDISC-SDTM	3.1.2	referencemetadata	column	refmeta	libref	input	dataset	N		.	
CDISC-SDTM	3.1.2	referencemetadata	table	refmeta	libref	input	dataset	N		.	
CDISC-SDTM	3.1.2	results	validationmetrics	results	libref	output	dataset	Y	&studyOutputPath/results	.	validation_metrics.sas7bdat
CDISC-SDTM	3.1.2	results	validationresults	results	libref	output	dataset	Y	&studyOutputPath/results	.	validation_results.sas7bdat
CDISC-SDTM	3.1.2	sourcedata		srcdata	libref	input	folder	N	&studyRootPath/data	.	
CDISC-SDTM	3.1.2	sourcemetadata	column	srcmeta	libref	input	dataset	N	&studyRootPath/metadata	.	source_columns.sas7bdat
CDISC-SDTM	3.1.2	sourcemetadata	table	srcmeta	libref	input	dataset	N	&studyRootPath/metadata	.	source_tables.sas7bdat
CDISC-SDTM	3.1.2	template		tmplt	libref	input	folder	N		1	
CDISC-TERMINOLOGY	NCI_THESAURUS	fmtsearch		ctfmt	libref	input	catalog	N	&_cstGRoot/standards/cdisc-terminology	2	cterms.sas7bcat
CST-FRAMEWORK	1.2	messages		cstmsg	libref	input	dataset	N	&_cstGRoot/standards/cst-framework-1.	1	messages.sas7bdat
CST-FRAMEWORK	1.2	template		csttmplt	libref	input	folder	N		2	

Here are some general observations about the sample SASReferences data set shown above:

- The **type** and **subtype** values are used by the SAS Clinical Standards Toolkit code to find the fileref or libref associated with a particular input or output file type.
- The SASref can be any user-defined value.
- An empty path tells the SAS Clinical Standards Toolkit to look for the default value from the metadata for a specific standard in the global standards library (usually a global standards library location).

A complete discussion of the SASReferences data set is provided in Chapter 5, “SASReferences File,” in the *SAS Clinical Standards Toolkit: User's Guide*. The SAS Clinical Standards Toolkit defines a standard SASReferences data set for each supported standard (found in the ***`global standards library directory/standards/standard/control`*** directory).

Using the SAS Clinical Standards Toolkit

The SAS Clinical Standards Toolkit global standards library contains the SAS metadata definition of all supported standards. This metadata definition, defined primarily as the reference_tables and reference_columns data sets, can be used as defined. The metadata definition can be modified or used as a template to build your own SAS representation of the CDISC standards provided by SAS or your own customized standard.

Sample driver modules are provided with each supported standard. These modules can be copied and modified to reflect your own data and metadata sources and the target location for any SAS process output. These drivers all follow the same general process workflow:

- Set any process global macro variable values.
- Define a root path for input and output files (unnecessary if you do not use relative paths and you specify explicit paths).
- Create or reference a SASReferences data set that defines all input/output files.
- Call the cstutil_processsetup() macro that confirms a valid SASReferences structure, allocates any SAS librefs and filerefs, sets macro autocall for a specific standard, and formats search paths.
- Call the primary macro of interest (such as validation or define creation macro).
- (Optional) Perform any session cleanup.

For a more complete description of what a driver module does, see “Running a Validation Process” in Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

2

Sample Scenarios

Validating the CDISC ADaM Data Sets

Assumption

You have a library of CDISC ADaM SAS data sets (for purposes of this example, derived from a library of CDISC SDTM 3.1.2 domains). Derivation of ADaM analysis files from SDTM domains is not a supported function in the SAS Clinical Standards Toolkit. Products such as SAS Clinical Data Integration can be used to create such mapping processes and transformation processes.

Location of ADaM Driver Modules

The ADaM driver modules are located in this directory:

***sample study library directory*/cdisc-adam-2.1-1.5/sascstdemodata/programs**

Step 1: Derive Metadata about Your Source Data

Before you can validate your ADaM data sets, you must derive a set of metadata that describes your library of analysis data sets. In the SAS Clinical Standards Toolkit, the metadata that describes your study data sets and columns is generally referred to as *source metadata*. To help derive this metadata, the SAS Clinical Standards Toolkit provides a sample driver module (create_sourcemetadata.sas) that calls the SAS macro adamutil_createsrcmetafromsaslib.sas found in the ***global standards library directory*/standards/cdisc-adam-2.1-1.5/macros** directory. This macro uses Base SAS metadata (proc contents output) and reference metadata (provided by SAS) describing the CDISC ADaM standard to initialize the source metadata. You might

find it necessary to augment or modify this approach based on other metadata that you have available or other processes that you adopt.

Step 2: Build Your Own Driver Module

The sample driver that can be run to demonstrate the ADaM validation process is validate_data.sas. Use this driver as a sample to build your own driver module, specifying the locations of source data and metadata. Take note of the SASReferences data set created in the validate_data driver. It references both SDTM and ADaM data and metadata as well as ADaM controlled terminology. Reference to the SDTM metadata supports comparison of ADaM column metadata with SDTM column metadata for those columns derived directly from SDTM. (See "Assumption" on page 10.)

Step 3: Submit the Modified Driver Module

The SAS Clinical Standards Toolkit validation processes generally create two types of output data sets: validation results and validation metrics. The names and locations of these SAS data sets depend on your SASReferences specifications for results management.

Here is a sample Results data set produced by the validation process:

Figure 2.1 *Partial Sample Results Data Set (CDISC ADaM 2.1 Validation Process)*

resultid	checkid	seqno	srcdata	message	resultseverity	resultflag	actual	keyvalues
CST0100	ADAM0001	1	SRCDATA.ADSL	No errors detected in SRCDATA.ADSL	Info	0		
CST0100	ADAM0048	1	SRCDATA.ADSL	No errors detected in source data	Info	0		
ADAM0053	ADAM0053	1	SRCDATA.ADAE	The values of USUBJID are not present in SDTM.DM	Error	1	STUDYID=SASCSTDEMODATA ,USUBJID=S999P999	USUBJID=S999P999, AETERM=, AESTDY=.
ADAM0053	ADAM0053	2	SRCDATA.ADQS	The values of USUBJID are not present in SDTM.DM	Error	1	STUDYID=SASCSTDEMODATA ,USUBJID=S999P999	USUBJID=S999P999, PARAM=.
ADAM0053	ADAM0053	3	SRCDATA.ADSL	The values of USUBJID are not present in SDTM.DM	Error	1	STUDYID=SASCSTDEMODATA ,USUBJID=S999P999	USUBJID=S999P999
ADAM0054	ADAM0054	1	SRCDATA.ADSL	Within ADSL there is more than one record for a unique value of USUBJID	Error	1	keys=USUBJID	USUBJID=S999P999
ADAM0061	ADAM0061	1	ADSL.TRTSDT+TRTSDTM	SDTM.EX is present and neither TRTSDT or TRTSDTM are present	Error	1		
ADAM0069	ADAM0069	1	SRCDATA.ADSL	A variable with a prefix of TR and containing AG is present and a variable with the same root with a suffix of N is not present	Error	1	TRAG1N	
CST0021	ADAM0102	1	CSTCHECK_COLUMN VARLIST	Table SRCDATA.ADQS does not contain APERIOD column(s)	Warning: Check not run	-1		
ADAM0102	ADAM0102	108	SRCDATA.ADAE	For every unique xx value of APERIOD in BDS datasets, there is not a ADSL variable TRTxxP	Error	1		
ADAM0138	ADAM0138	1	SRCDATA.ADAE	CRITy is populated and CRITyFL is not populated	Error	1	CRIT1=2,CRIT1FL=	USUBJID=S999P999, AETERM=HEARTBURN-LIKE DYSPEPSIA, AESTDY=3
ADAM0143	ADAM0143	2	SRCDATA.ADQS	PARAMCD has more than 8 characters in length	Error	1	PARAMCD=leucocytes	USUBJID=S999P999, PARAM=lcount

The validation results shown above are representative of the range of validation results one might see, from no reported errors (such as ADAM0001) to multiple errors detected (such as ADAM0053) to an inability to run a specific check because of a lack of data or metadata (such as ADAM0102). The validation metrics output data set attempts to summarize the validation results and provide a denominator for each check.

For a more thorough discussion of how validation is performed, see Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Creating a define.xml File from CDISC SDTM 3.1.3 Source Data

Overview

The SAS Clinical Standards Toolkit supports the currently published CDISC CRT-DDS 1.0 (define.xml) submission standard, which supports representation of the CDISC SDTM 3.1.1, 3.1.2, and 3.1.3 tabulation data sets and ADaM 2.1 analysis data sets in metadata form.

Assumption

You have a library of CDISC SDTM SAS data sets (which are not read by this process) from which a set of metadata (in the form of source_study, source_tables, source_columns, source_values, and source_documents data sets) has been created. This metadata must contain the expected, correctly typed columns created for the sample study provided by SAS.

Location of CRT-DDS Driver Modules

The CRT-DDS driver modules are located in this directory:

***sample study library directory*/cdisc-crtdds-1.0-1.5/programs**

Step 1: Extract Available SDTM Metadata into CRT-DDS Metadata Files

The initial task is to extract available SDTM metadata into CRT-DDS metadata files. The SAS representation of CRT-DDS involves 39 data sets, but only 19 of these derive directly from the SDTM metadata. The sample driver create_crtdds_from_sdtm.sas, modified to point to your specific SDTM study, should be submitted to extract the SDTM metadata. This process builds all 39 data sets but populates only 19 (depending on your study).

Note: The key input (source) files are the SDTM metadata files (source_study, source_tables, source_columns, source_values, and source_documents), not the SDTM domain data sets.

Step 2: (Optional) Populate the Remaining CRT-DDS Data Sets

Note: If you do not have additional metadata, you can skip this step.

None of the remaining 39 data sets must be populated to create a viable define.xml file using the SAS Clinical Standards Toolkit. However, they might be needed to further populate the remaining CRT-DDS data sets using your own metadata sources and methods.

For more discussion of this step, see FAQ “How Do I Add Supplemental Data (Not Directly Derivable from Other Standards Such as CDISC SDTM) Used to Create a define.xml File?” on page 32. For a discussion of the data sets most critical for successful derivation of the define.xml file, see “Special Topic: A Round Trip Exercise Involving the CDISC CRT-DDS Standard: Importing and Exporting the define.xml File” in Chapter 8, “XML-Based Standards,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Step 3: Create the define.xml File

At this point, all available content for the define.xml file has been captured in the SAS representation (39 data sets) of the CDISC CRT-DDS standard. The SAS Clinical Standards Toolkit provides a sample driver module, create_crtdds_define.sas. This module builds and validates the define.xml file. Submit the create_crtdds_define.sas driver module.

In this driver, the call to the primary task macro requests that the default style sheet provided by SAS (source: CDISC) be copied to the folder location containing the generated define.xml file. The macro is located in the ***global standards library directory*/standards/cdisc-crtdds-1.0-1.5/macros** directory.

Here is the macro:

```
%crtdds_write(_cstCreateDisplayStyleSheet=1);
```

Here is a portion of the define.xml file, as rendered by this default style sheet. Hyperlinks among tables, columns, codelists, and other file elements are provided.

Figure 2.2 *Partial Sample define.xml File (as Rendered by the Default Style Sheet)*

Datasets for Study study1

Dataset	Description	Structure	Purpose	Keys	Location
AE	Adverse Events	Events - One record per adverse event per subject	Tabulation	STUDYID USUBJID AEDECOD AESTDTC	Adverse Events SAS transport file
CE	Clinical Events	Events - One record per event per subject	Tabulation	STUDYID USUBJID CETERM CESTDTC	Clinical Events SAS transport file
CM	Concomitant Medications	Interventions - One record per recorded medication occurrence or constant-dosing interval per subject	Tabulation	STUDYID USUBJID CMTRT CMSTDTC	Concomitant Medications SAS transport file

Step 4: Validate the Generated define.xml File

The final step is to validate the generated define.xml file. As discussed in Chapter 8, “XML-Based Standards,” in the *SAS Clinical Standards Toolkit: User's Guide*, the SAS Clinical Standards Toolkit offers two complementary validation methodologies.

- The first methodology relies on the definition of a master set of validation checks as described in the previous example. This method uses SAS files and SAS code to

validate the SAS representation of the standard. Example checks include the assessment of foreign key relationships across data sets and value conformance to a set of expected values.

- The second methodology involves verification that the XML file is valid structurally and syntactically according to the XML schema.

The final task in the sample create_crtdds_define.sas driver is to call the crtdds_xmlvalidate() macro to perform the schema validation.

Here is a sample Results data set produced by the validation process:

Figure 2.3 *Partial Sample Results Data Set (CDISC CRT-DDS 1.0 Create Process)*

	resultid	resultseq	seqno	srcdata	message	resultseverity
1	CST0108	1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibraryTK15/standards/cst-framework-1.5/programs/initialize.p	Info
2	CST0200	1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to c:/cstGlobalLibraryTK15/standards/cst-framework-1.5/templates to perform the template lookup	Info
3	CST0102	1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences was created as requested	Info
4	CST0200	1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\FRJANS~1.CAR\AppData\Local\Temp\SAS Temporary Files_TD8928_L72371_/sasreferences	Info
5	CST0200	1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info
6	CST0200	1	2	CSTUTIL_ALLOCATESASREFERENCE	SASReferences data set was successfully validated	Info
7	CST0108	1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibraryTK15/standards/cdisc-crtdds-1.0-1.5/programs/initialize	Info
8	CST0200	1	1	CRTDDS_WRITE	PROCESS STANDARD: CDISC-CRTDDS	Info
9	CST0200	1	2	CRTDDS_WRITE	PROCESS STANDARDVERSION: 1.0	Info
10	CST0200	1	3	CRTDDS_WRITE	PROCESS DRIVER: CREATE_CRTDDS_DEFINE	Info
11	CST0200	1	4	CRTDDS_WRITE	PROCESS DATE: 2013-02-28T17:47:32	Info
12	CST0200	1	5	CRTDDS_WRITE	PROCESS TYPE: CREATE CRTDDS DEFINE.XML	Info
13	CST0200	1	6	CRTDDS_WRITE	PROCESS SASREFERENCES: C:\Users\FRJANS~1.CAR\AppData\Local\Temp\SAS Temporary Files_TD8928_L72371_/sasreferences	Info
14	CST0200	1	7	CRTDDS_WRITE	PROCESS STUDYROOTPATH: c:/cstSampleLibraryTK15/cdisc-crtdds-1.0-1.5	Info
15	CST0200	1	8	CRTDDS_WRITE	PROCESS GLOBALLIBRARY: c:/cstGlobalLibraryTK15	Info
16	CST0200	1	9	CRTDDS_WRITE	PROCESS CSTVERSION: 1.5	Info
17	CST0122	1	1	CST_CREATETABLESFORDATASTAN	The tables were created for CDISC-CRTDDS 1.0 in library _cst6809	Info
18	CST0200	1	1	JAVA CHECK	No Java issues	Info
19	CRT0001	1	1	XML TRANSFORMER	Transform starting.	Info
20	CRT0001	1	2	XML TRANSFORMER	Using JRE: C:\PROGRA~2\Java\JRE16~1.0_2	Info

The Results data set provides process information and the location of the generated define.xml file. The Results data set confirms that no problems were found with the file following validation of the file.

Note: The SAS validation of file content was not run in this example.

Importing and Validating a CDISC ODM XML File

Overview

The SAS Clinical Standards Toolkit supports the currently published CDISC ODM 1.3.0 and 1.3.1 standards, which facilitate the archival and interchange of the metadata and data for clinical research. The SAS Clinical Standards Toolkit can import an ODM XML file into a SAS data set representation of the ODM 1.3.0 or 1.3.1 standard.

Assumption

You have an ODM XML file (which is not created by this process) from which a SAS representation has to be created.

Location of ODM Driver Modules

The ODM driver modules are located in this directory:

***sample study library directory*/cdisc-odm-1.3.1-1.5/programs**

Step 1: Validate the ODM XML File against the ODM 1.3.1 XML Schema

The initial task is to validate the ODM XML file against the ODM 1.3.1 XML schema. This involves verifying that the ODM XML file is valid structurally and syntactically according to the XML schema. The sample driver create_sasodm_fromxml.sas, modified to point to your specific ODM XML file, contains a call to the odm_xmlvalidate macro to perform validation.

Here is a sample Results data set produced by the schema validation process:

Figure 2.4 *Partial Sample Results Data Set (CDISC ODM 1.3.1 Schema Validation Process)*

	resultid	seqno	srcdata	message	resultseverity
5	CST0200	1	ODM_XMLVALIDATE	PROCESS STANDARD: CDISC-ODM	Info
6	CST0200	2	ODM_XMLVALIDATE	PROCESS STANDARDVERSION: 1.3.1	Info
7	CST0200	3	ODM_XMLVALIDATE	PROCESS DRIVER: CREATE_ODMXML	Info
8	CST0200	4	ODM_XMLVALIDATE	PROCESS DATE: 2011-11-27T14:55:01	Info
9	CST0200	5	ODM_XMLVALIDATE	PROCESS TYPE: VALIDATE ODM XML	Info
10	CST0200	6	ODM_XMLVALIDATE	PROCESS SASREFERENCES: C:\Users\frjans\AppData\Local\Temp\SAS Temporary Files_TD1328_L72371_/_cstsasrefs.sas7bdat	Info
11	CST0200	7	ODM_XMLVALIDATE	PROCESS STUDYROOTPATH: ..	Info
12	CST0200	8	ODM_XMLVALIDATE	PROCESS GLOBALLIBRARY: c:/cstGlobalLibraryTK15	Info
13	CST0200	9	ODM_XMLVALIDATE	PROCESS CSTVERSION: 1.5	Info
14	CST0200	1	JAVA CHECK	No Java issues	Info
15	ODM0001	1	XML TRANSFORMER	Transform starting.	Info
16	ODM0001	2	XML TRANSFORMER	Using JRE: C:\PROGRA~2\Java\jre6	Info
17	ODM0001	3	XML TRANSFORMER PARAMETER	Import Or Export: EXPORT	Info
18	ODM0001	4	XML TRANSFORMER PARAMETER	Standards XML Path: C:/SASPlaypen/TK1.5/cdisc-odm-1.3.1/sasmisc/sample/cdisc-odm-1.3.1/sourcexml/odm_sample.xml	Info
19	ODM0001	5	XML TRANSFORMER PARAMETER	Fail on Validation Error: false	Info
20	ODM0001	6	XML TRANSFORMER PARAMETER	Standard Name: CDISC-ODM	Info
21	ODM0001	7	XML TRANSFORMER PARAMETER	Standard Version: 1.3.1	Info
22	ODM0001	8	XML TRANSFORMER PARAMETER	Schema Repository Location: c:/cstGlobalLibraryTK15/schema-repository	Info
23	ODM0001	9	XML TRANSFORMER PARAMETER	XSL Repository Location: null	Info
24	ODM0001	10	XML TRANSFORMER PARAMETER	Output Encoding: UTF-8	Info
25	ODM0001	11	XML TRANSFORMER PARAMETER	Log File Location: C:\Users\frjans\AppData\Local\Temp\SAS Temporary Files_TD1328_L72371_/_log0878	Info
26	ODM0001	12	XML TRANSFORMER PARAMETER	Header Comment Text: Produced from SAS data using the SAS Clinical Standards Toolkit	Info
27	ODM0001	13	XML TRANSFORMER PARAMETER	Is Validating XML: true	Info
28	ODM0001	14	XML TRANSFORMER PARAMETER	Creating Display Stylesheet: false	Info
29	ODM0001	15	XML TRANSFORMER PARAMETER	Custom Stylesheet: null	Info
30	ODM0001	16	XML TRANSFORMER PARAMETER	Custom Stylesheet Output Shortname: null	Info
31	ODM0001	17	XML TRANSFORMER PARAMETER	Creating Output Folders: true	Info
32	ODM0001	18	XML TRANSFORMER	The document validated successfully	Info
33	ODM0115	1	ODM_XMLVALIDATE	No errors were found in the ODM file.	Info

This Results data set provides process information and the location of the ODM XML file, and confirms that the ODM XML file validated successfully.

Here is a sample Results data set that shows schema validation issues with an ODM XML file:

Figure 2.5 *Partial Sample Results Data Set (CDISC ODM 1.3.1 Schema Validation Process) – Invalid ODM XML File*

	resultid	seqno	srcdata	message	resultseverity
32	ODM0003	18	XML VALIDATION	(Line 104/Column 152) cvc-enumeration-valid: Value 'N' is not facet-valid with respect to enumeration '[Yes, No]'. It must be a value from the enumeration.	Error
33	ODM0003	19	XML VALIDATION	(Line 104/Column 152) cvc-attribute.3: The value 'N' of attribute 'Repeating' on element 'ItemGroupDef' is not valid with respect to its type, 'YesOrNo'.	Error
34	ODM0003	20	XML VALIDATION	(Line 512/Column 15) The element type "ItemDef" must be terminated by the matching end-tag "</ItemDef>".	Error
35	ODM0003	21	XML TRANSFORMER	The element type "ItemDef" must be terminated by the matching end-tag "</ItemDef>".	Error
36	ODM0011	1	ODM_XMLVALIDATE	Errors were reported in the generation of the ODM file.	Error

Running the odm_xmlvalidate macro is not required to be able to import an ODM XML file. However, importing an invalid ODM XML file can result in an incomplete import because the odm_read macro ignores elements and attributes in the ODM XML file that are not defined by the XML schema.

Step 2: Import the ODM XML File

The next step is to import the ODM XML file by calling the odm_read macro, which results in 76 SAS data sets describing the CDISC ODM 1.3.1 data model. The odm_read macro is called by the sample driver create_sasodm_fromxml.sas.

Here is a sample Results data set. It provides process information and the location of the imported ODM XML file. The Results data set confirms that no problems were found

with the import. And, the Results data set shows that format catalogs and data sets have been created.

Figure 2.6 *Partial Sample Results Data Set (CDISC ODM 1.3.1 Read Process)*

	resultid	seqno	srcdata	message	resultseverity
8	CST0200	1	ODM_READ	PROCESS STANDARD: CDISC-ODM	Info
9	CST0200	2	ODM_READ	PROCESS STANDARDVERSION: 1.3.1	Info
10	CST0200	3	ODM_READ	PROCESS DRIVER: CREATE_SASODM_FROMXML	Info
11	CST0200	4	ODM_READ	PROCESS DATE: 2013-03-27T16:51:26	Info
12	CST0200	5	ODM_READ	PROCESS TYPE: FILEIO	Info
13	CST0200	6	ODM_READ	PROCESS SASREFERENCES: C:\Users\FRJANS~1.CAR\AppData\Local\Temp\SAS Temporary Files_TD7264_L72371_/_cstsasrefs.sas7bdat	Info
14	CST0200	7	ODM_READ	PROCESS STUDYROOTPATH: c:\cstSampleLibrary/cdisc-odm-1.3.1-1.5	Info
15	CST0200	8	ODM_READ	PROCESS GLOBALLIBRARY: c:\cstGlobalLibrary	Info
16	CST0200	9	ODM_READ	PROCESS CSTVERSION: 1.5	Info
17	CST0200	1	JAVA CHECK	No Java issues	Info
18	ODM0013	1	ODM_READ	The ODM map file was read from the following location: c:\cstSampleLibrary\cdisc-odm-1.3.1-1.5\referencexml\odm.map	Info
19	CST0200	2	ODM_READ	Destination library for format catalogs set to work	Info
20	CST0200	3	CSTUTIL_BUILDFORMATSFROMXML	work.ODMfmtcat_de catalog and data set created	Info
21	CST0200	4	CSTUTIL_BUILDFORMATSFROMXML	work.ODMfmtcat_en catalog and data set created	Info
22	CST0200	5	CSTUTIL_BUILDFORMATSFROMXML	work.ODMfmtcat_fr_CA catalog and data set created	Info
23	ODM0012	6	ODM_READ	The ODM file c:\cstSampleLibrary\cdisc-odm-1.3.1-1.5\sourcexml\odm_sample.xml was read successfully.	Info
24	CST0102	1	CSTUTIL_SAVERESULTS	results.read_results was created as requested	Info

Step 3: Validate the SAS Representation of the ODM XML File

The next step is to validate the SAS representation of the ODM XML standard. In step 1, the ODM XML file was validated against the XML schema. Validating the SAS representation of the ODM XML standard goes further than XML schema validation.

An example of a check that is part of the validation of the SAS representation is the assessment of foreign key relationships across data sets. The SAS Clinical Standards Toolkit provides a sample driver program, validate_odm_data.sas, which validates the SAS representation of an ODM XML file.

Submit the modified driver module. The SAS Clinical Standards Toolkit validation processes generally create two output files: validation results and validation metrics. The names and locations of these SAS data sets depend on your SASReferences specifications for results management.

Here is a sample Results data set produced by the validation process:

Figure 2.7 *Partial Sample Results Data Set (CDISC ODM 1.3.1 Validation Process)*

	resultid	seqno	srcdata	message	resultseverity	actual	keyvalues
197	ODM0110	1	SRCDATA.ITEMDATA (SRCDATA.ITEMDEFS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDATA	Error	ITEMOID=ID.AETERM	
198	ODM0110	2	SRCDATA.ITEMDATA (SRCDATA.ITEMDEFS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDATA	Error	ITEMOID=ID.AETERM	
199	CST0100	1	SRCDATA.ITEMDATA (SRCDATA.ITEMGROUPDATA)	No errors detected in source data	Info		
200	CST0100	1	SRCDATA.ITEMDATA (SRCDATA.MEASUREMENTUNITS)	No errors detected in source data	Info		
201	CST0100	1	SRCDATA.ITEMDATA (SRCDATA.SIGNATURE)	No errors detected in source data	Info		
202	ODM0110	1	SRCDATA.ITEMDEFS (SRCDATA.CODELISTS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDEFS	Error	CODELISTREF=CodeLists.OID.UNIT	FK_METADATAVERSION=MetaDataVersion.OID.1, OID=ItemDef.OID.LB.LBORRESU
203	ODM0110	2	SRCDATA.ITEMDEFS (SRCDATA.CODELISTS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDEFS	Error	CODELISTREF=CodeLists.OID.UNIT	FK_METADATAVERSION=MetaDataVersion.OID.1, OID=ItemDef.OID.LB.LBSTRESU
204	ODM0110	3	SRCDATA.ITEMDEFS (SRCDATA.CODELISTS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDEFS	Error	CODELISTREF=CodeLists.OID.LBTEST	FK_METADATAVERSION=MetaDataVersion.OID.1, OID=ItemDef.OID.LB.LBTEST
205	ODM0110	4	SRCDATA.ITEMDEFS (SRCDATA.CODELISTS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDEFS	Error	CODELISTREF=CodeLists.OID.UNIT	FK_METADATAVERSION=MetaDataVersion.OID.1, OID=ItemDef.OID.lnr.lborresu
206	CST0100	1	SRCDATA.ITEMDEFS (SRCDATA.METADATAVERSION)	No errors detected in source data	Info		

In this example, there are several issues in the foreign key relationships across data sets.

Step 4: Extract Domain Data from the ODM XML File

The final step is to extract domain data from the ODM XML file by calling the odm_extractdomaindata macro. The odm_extractdomaindata macro extracts one or more data sets from the ClinicalData or ReferenceData sections of the ODM XML file.

The odm_extractdomaindata macro is called by the sample driver extract_domaindata.sas (extract one domain) or by the sample driver extract_domaindata_all.sas (extract all domains).

Here is an example of the extracted AE data set:

Figure 2.8 *Example Extracted AE Data Set*

	__StudyOID	__MetaDataVersionOID	__SubjectKey	__StudyEventOID	__StudyEventRepeatKey	__FormOID	__FormRepeatKey
1	Study.OID	MetaDataVersion.OID.1	S001P011	StudyEventDefs.OID.6.AdverseEvent	1	FormDefs.OID.AE	1
2	Study.OID	MetaDataVersion.OID.1	S001P011	StudyEventDefs.OID.6.AdverseEvent	1	FormDefs.OID.AE	1

	__ItemGroupOID	__ItemGroupRepeatKey	__TransactionType	__LocationOID	__UserOID	TAREA	PNO	SCTRY	F_STATUS
1	ItemGroupDefs.OID.AE	1	Insert	Location.OID.S001	User.OID.I008	Oncology	143-02	United States	Source verified, queried
2	ItemGroupDefs.OID.AE	2	Insert	Location.OID.S001	User.OID.I008	Oncology	143-02	United States	Source verified, queried

	LINE_NO	AETERM	AESTMON	AESTDAY	AESTYR	AESTDT	AEENMON
1	1	HEADACHE	6	10	1999	1999-06-10	6
2	2	CONGESTION	6	11	1999	1999-06-11	6

	AEENDAY	AEENYR	AEENDT	AESEV	AEREL	AEOUT	AEACTTRT	AECONTRT
1	14	1999	1999-06-14	Mild	None	Resolved, no residual effects	None	Medication required
2	14	1999	1999	Mild	None	Continuing	None	Medication required

3

Frequently Asked Questions

Global Standards Library

Does It Matter Where the Global Standards Library is Located?

No. On Microsoft Windows systems, the default location for the global standards library is the **`c:/cstGlobalLibrary`** directory. On UNIX systems, the default location for the global standards library is the **`/usr/local/cstGlobalLibrary`** directory.

This location is assumed (and referenced as *global standards library directory*) throughout this document. *global standards library directory* can point to any location that is accessible by SAS.

For a summary of the cstGlobalLibrary, see “Global Standards Library” in Chapter 2, “Framework,” in the *SAS Clinical Standards Toolkit: User's Guide*.

How Do I Change the Location of the Global Standards Library?

The value of the variable cstGlobalLibrary is set during installation of the SAS Clinical Standards Toolkit. If you want to change the location after installation, you must modify the framework autocall utility macro cstutil_setcstgroot to point to the new location that you have chosen.

Note: The SAS Clinical Standards Toolkit recognizes and uses only one global standards library at a time.

Process Results

How Do I See Process Results?

The default location for the results of a SAS Clinical Standards Toolkit process is set by the _cstResultsDS property in the framework initialize.properties file. By default, this is set to work._cstresults. Process results can be persisted, if the SASReferences data set has a type=results record.

Note: All SAS Clinical Standards Toolkit process Results data sets have the same structure and all columns have the same meaning.

Standards

How Do I Customize a Registered Standard?

The definition of each standard (and version of each standard) is done in the global standards library. A best practice recommendation is that you do not (permanently) modify the global standards library files, but instead modify copies of the files.

The files and metadata that constitute a standard vary from standard to standard. Use the files that define each standard, located in the ***global standards library directory*/standards/*standard*** directory, as templates, adding your customizations. You can rename or otherwise introduce your own subfolders and files that constitute each standard. If you want to create your own version of an existing registered standard, review the information in the next question.

How Do I Register a New Standard (or a New Version for an Existing Standard)?

Use the SAS Clinical Standards Toolkit framework utility macro cst_registerstandard.sas to register a new standard. A new subfolder hierarchy is added to the ***global standards library directory*****/standards/** directory, and the ***global standards library directory*****/metadata/standards.sas7bdat** data set is updated with metadata about the new standard. This metadata is provided in the form of two data sets, standards, and StandardSASReferences. These data sets are passed to the SAS Clinical Standards Toolkit as parameters in the call to the cst_registerstandard macro.

See “Maintenance Usage Scenarios” in Chapter 2, “Framework,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Does SAS Offer Updates for Previously Released Standards?

New production releases of the SAS Clinical Standards Toolkit routinely provide updates to previously released standards. For information about how these updates are provided, see “Does a New Version of the SAS Clinical Standards Toolkit Automatically Overwrite Files Installed with a Previous Version?” on page 39.

SAS also makes available a SAS support knowledge base focus area (**http://support.sas.com/rnd/base/cdisc/cst/index.html**) to communicate product information, references, and updates. Preproduction files are also available for download.

The SAS Clinical Standards Toolkit development team also supports the early adopter phase and works with development partners on future product releases.

Creating and Modifying Files

Should I Modify SAS Clinical Standards Toolkit Files Supplied by SAS If I Want to Test Changes or Make Permanent Changes to Any of Those Files?

A best practice recommendation is that you do not (permanently) modify framework autocall macros or global standards library files, but instead modify copies of the files. This allows seamless updates to those files provided by SAS without concern about overwriting or losing your changes.

Note: The SAS Clinical Standards Toolkit provides an internal backup of the global standards library files for specific standards provided by SAS in the *sample study library directory* subfolders for a specific standard.

How Do I Modify a Sample Driver Program to Point to My Own Study Data?

Each sample driver module sets a &studyRootPath macro variable to point to the root path of the sample study SAS supplies for each standard. This macro enables you to point to your own study root path by resetting the value of this macro variable. You might instead prefer to use explicit paths in your SASReferences data set and not use &studyRootPath at all.

The sample drivers assume a study folder hierarchy like that of the sample study, but this hierarchy is not required. If you have another organization for your study, you might need to make other changes to the paths specified in your driver module.

How Do I Create the Required Study Metadata?

Each supported standard might define study metadata differently. The SAS representation of the 32 standard CDISC SDTM 3.1.2 domains differs from the SAS representation of the three types of tables supported in CDISC ADaM 2.1.

Most users have metadata about their studies in some form and file format other than that used by the SAS Clinical Standards Toolkit. You have to convert or map some of that metadata into the source_study, source_tables, and source_columns format specified in the sample study metadata folder (such as the ***`sample study library directory`*****`/cdisc-adam-2.1-1.5/sascstdemodata/metadata`** directory). Some standards provide sample code that supports this task. For CDISC ADaM, look at the create_sourcemetadata.sas sample driver module as an example.

Note: The structure of the source metadata corresponds to the reference metadata for each standard as defined in the global standards library (such as ***`global standards library directory`*****`/standards/cdisc-adam-2.1-1.5/metadata`** reference_tables and reference_columns data sets).

Can I Modify the Check Metadata Columns tableScope and columnScope to Match My Study?

Yes. The initial values are based on the sample data associated with each standard. Your study data almost certainly differs. For CDISC SDTM and CDISC ADaM, your domains and analysis data sets reflect your study protocol. Some checks might not apply to some of your domains because your implementation of that domain might deviate from other domains from the same class (such as Findings).

Be aware that tableScope and columnScope wildcarding is a convenience, and you can instead explicitly reference a specific table or column in any check invocation.

See “Validation Check Metadata: Validation Master” in Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Can I Add My Own New Validation Checks for SDTM or ADaM?

Yes. You can add checks to any SAS Clinical Standards Toolkit standard that supports validation. Each check is defined as a set of metadata (in the validation_master data set found in the ***global standards library directory*****/standards/*****standard*****/validation/control** directory) and a message to be used when an error is detected (in the messages data set found in the ***global standards library directory*****/standards/*****standard*****/messages** directory).

For each new check, you must create the check metadata to conform to the validation_master template and an associated message to conform to the messages data set metadata structure.

For more information about check metadata and the available SAS Clinical Standards Toolkit validation check macros, see Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

How Do I Use My Own Controlled Terminology?

The SAS Clinical Standards Toolkit permits the use of any set of controlled terminology or any coding dictionaries. Generally, controlled terminology is defined to the SAS Clinical Standards Toolkit as SAS format catalogs; coding dictionaries as SAS data sets. Either format is valid.

The SASReferences data set is used to document these format catalogs and coding dictionaries, and to facilitate run-time references to these reference input sources. In the SAS Clinical Standards Toolkit sample drivers, a SASReferences type=fmtsearch record points to each SAS format catalog (and allows specification of a reference order for like-named formats), and a type=referencecterm record points to each specific coding dictionary to be referenced. The format search path is set with the call to the cstutil_processsetup macro.

For more information about using controlled terminology in validation processes, see “Special Topic: Using Alternative Controlled Terminologies” in Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

How Do I Point to My Company’s Version of MedDRA?

The sample SDTM SASReferences data set offers an example of referencing the MedDRA dictionary. The type=referencecterm record points to a folder (which can be anywhere) that can contain a MedDRA data set (using any name).

The SDTM validation check (SDTM0451) provided by SAS that compares AEDECOD to MedDRA preferred terms expects that the MedDRA data set contains the column PT_NAME. If your data set uses other column names, see “If I Disagree with or Choose to Modify the Codelogic of a Specific Validation Check, What Needs to Be Done?” on page 36.

See “Building a Validation Process” in Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

How Do I Use My Own Style Sheet with a SAS Clinical Standards Toolkit-Generated define.xml File?

The SAS Clinical Standards Toolkit uses the macro crtdds_write to create a define.xml file. This macro has the optional parameter _cstCreateDisplayStyleSheet that determines two things:

- whether the macro creates a style sheet in the same folder as the output define.xml file
- whether the macro creates a reference to that style sheet in the define.xml file

If you set this macro parameter to `1`, the macro looks in the provided SASReferences file for a record with a type/subtype of **`referencexml/stylesheet`** and uses that file.

By default, the crtdds_write macro creates a style sheet reference in the define.xml file and copies a style sheet in the same folder as the define.xml file. This default style

sheet, define1-0-0.xsl, is copied from the ***global standards library directory*****/standards/cdisc-crtdds-1.0/stylesheet** directory and is identical to the style sheet that was originally made available by CDISC as part of the first release of the define.xml file.

See “Writing XML Files” in Chapter 8, “XML-Based Standards,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Can I Associate a Style Sheet with an ODM XML File Generated by the SAS Clinical Standards Toolkit?

Although it is not common to associate a style sheet with an ODM XML file, the SAS Clinical Standards Toolkit allows this. To do this, you must call the macro odm_write with macro parameter cstCreateDisplayStyleSheet=1 and also add a record to the SASReferences file with type/subtype of **referencexml/stylesheet** pointing to the style sheet to be used.

Note: SAS does not supply a default style sheet for ODM.

See the online macro API reference material regarding ODM.

How Do I Add Supplemental Data (Not Directly Derivable from Other Standards Such as CDISC SDTM) Used to Create a define.xml File?

When the SAS Clinical Standards Toolkit creates a define.xml file, it converts the information from a SAS data set representation of the CRT-DDS model into XML. For CDISC CRT-DDS 1.0, this means that 39 data sets (such as ItemDefs) are the source for creating the define.xml element and attribute structure. The content of these 39 data sets can be derived in part from other standards (such as CDISC SDTM).

You can directly maintain these 39 data sets so that you can add additional data that needs to be part of the define.xml. To be able to maintain these 39 data sets, you must be familiar with the structure of these data sets and the relationships among these data

sets. For example, to add Value Level Metadata to the define.xml, add rows to four data sets: itemdefs, valuelists, valuelistitemrefs, and itemvaluelistrefs.

See the online API reference material regarding CRT-DDS 1.0 SAS data sets and "Writing XML Files" in Chapter 8, "XML-Based Standards," in the *SAS Clinical Standards Toolkit: User's Guide*.

How Do I Create the Data Sets That Are Used to Create an ODM XML File?

When the SAS Clinical Standards Toolkit creates an ODM XML file, it converts the information from a SAS data set representation of the ODM model into XML. For CDISC ODM 1.3.1, this means that 76 data sets (such as ItemDefs) are the source for creating the ODM XML element and attribute structure.

You can directly maintain these 76 data sets so that you can add data that needs to be part of the ODM XML file. To be able to maintain these 76 data sets, you must be familiar with the structure of these data sets and the relationships among these data sets.

Normally, you would not use SAS to create an ODM XML file, because it is assumed that there is a workflow in which a data collection and management system exports metadata definitions and data content in the form of a transactional or snapshot ODM XML file. When SAS is used to create an ODM XML file, the initial SAS representation of the ODM model can be created by using one of the following methods:

- Call %cst_createTablesForDataStandard(_cstStandard=CDISC-ODM, _cstStandardVersion=1.3.1, _cstOutputLibrary=work); to build zero-observation SAS data sets.
- Call the %odm_read macro to import an ODM XML file that can serve as a template for the creation of the SAS representation of the ODM XML data model.

See "Writing XML Files" in Chapter 8, "XML-Based Standards," in the *SAS Clinical Standards Toolkit: User's Guide*.

How Do I Modify the autocall Path to Point to My Own Macros?

The SAS Clinical Standards Toolkit autocall path is set by default to `SASAUTOS`, which includes the SAS Clinical Standards Toolkit framework macros found in the `!sasroot\cstframework\sasmacro` directory (Microsoft Windows) or in the `!sasroot/sasautos` directory (UNIX).

A macro library is defined for each standard in the global standards library. (See "Where Do the Framework and Standard-Specific Macros Reside?" on page 37.) In the SASReferences data set, one or more records with type=autocall can be used to reset the SAS autocall path, where the order column specifies the order each macro library is to be referenced. You can use the same strategy to reference any user-defined macro libraries, even if those libraries are not associated with a specific standard.

See Chapter 5, "SASReferences File," in the *SAS Clinical Standards Toolkit: User's Guide*.

Do I Have to Create and Use a SASReferences Data Set for Any SAS Clinical Standards Toolkit Process That I Run?

No. However, the SAS Clinical Standards Toolkit has adopted this method to define and document the input and output file and library references, to build the macro autocall, and to format search paths for each SAS Clinical Standards Toolkit process. Although you can perform all library and file allocations and set SAS system options yourself, it is highly recommended that you use the SASReferences approach.

See Chapter 5, "SASReferences File," in the *SAS Clinical Standards Toolkit: User's Guide*.

Do I Need to Use Properties Files to Define My Own Global Macro Variables?

No. However, doing so standardizes your approach to creating and documenting global macro variables.

Within the SAS Clinical Standards Toolkit, either of these two methods is always used to process properties files:

- adding a type=properties record to the SASReferences data set, which is then processed in the routine call to cstutil_processsetup()
- directly calling cst_setproperties()

See “Properties” in Chapter 3, “Metadata File Descriptions,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Validation Process

Do I Have to Run All of the SAS Validation Checks for SDTM?

No. The SAS Clinical Standards Toolkit offers the full set of CDISC SDTM 3.1.2 validation checks in the global standards library data set called validation_master found in the ***`global standards library directory`*`/standards/cdisc-sdtm-3.1.2-1.5/validation/control`** directory.

This set of checks represents an amalgamation of WebSDM, OpenCDISC, and SAS checks. Some checks assess structural (metadata) compliance, and other checks evaluate specific data sets, columns, and data values at the “cell” level.

For any given validation process, the SAS Clinical Standards Toolkit expects that you have defined and referenced (in SASReferences, with type/subtype set to **`control/validation`**) a run-time set of checks that might be some subset of the validation_master checks. You have the option of both creating this subset and ordering the checks to be run.

The run-time set of checks might well change over time. For example, once the metadata structure of the domains is confirmed, those checks do not need to be run again.

See “Building a Validation Process” in Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Can I Run a Validation Process in the SAS Clinical Standards Toolkit on Multiple Studies Simultaneously?

Yes. The SAS Clinical Standards Toolkit offers several distinct methodologies that support validating multiple studies. However, these methodologies depend on your data accrual processes and workflow. It is important that you maintain relationships between the source study data and the metadata that the SAS Clinical Standards Toolkit requires. Each column in each data set in each study must be uniquely identifiable.

See “Case Study 7: Validation of Multiple Studies” in Chapter 6, “Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

If I Disagree with or Choose to Modify the Codelogic of a Specific Validation Check, What Needs to Be Done?

Create your own copy of the validation_master data set for a specific standard (which might be the run-time validation_control data set). Then, make the changes to the copy’s validation check metadata, including codelogic.

If you believe that the codelogic provided by SAS is in error and you want SAS to correct the error, report the problem through normal Technical Support channels. Approved updates become available with a subsequent hot fix or release of the SAS Clinical Standards Toolkit.

Macros

Is There Any Debugging Help for SAS Macro Errors?

Debugging errors in SAS macros can be difficult. The SAS Clinical Standards Toolkit defines the global macro variable_cstDebug to enable these SAS system options: mprint, mlogic, symbolgen, and mautolocdisplay.

Use these options to significantly increase the number of lines written to the SAS log. By default, _cstDebug is set to 0, and these options are not enabled.

Common errors might be discussed in various parts of the *SAS Clinical Standards Toolkit: User's Guide*. SAS also maintains a knowledge base that includes notes regarding installation, problems, and usage for specific products. (See "How Do I Report Problems, Ask Questions, or Get More Information about How to Use the SAS Clinical Standards Toolkit?" on page 38.)

Other users might have reported or addressed problems on the SAS and Clinical Trials Community forum (`http://communities.sas.com/community/sas_and_clinical_trials`).

See "Common Errors and Solutions" in Chapter 5, "SASReferences File," in the *SAS Clinical Standards Toolkit: User's Guide*.

Where Do the Framework and Standard-Specific Macros Reside?

The SAS Clinical Standards Toolkit framework macros can be found here:

- Microsoft Windows

  `!sasroot\cstframework\sasmacro`, where !sasroot is `C:\Program Files\SASHome\SASFoundation\9.3`

- UNIX

!sasroot/sasautos, where !sasroot is **/usr/local/SASHome/SASFoundation/9.3/**

Macros for specific standards can be found in the ***global standards library directory*/standards/*standard*/macros** directory.

Miscellaneous

How Do I Report Problems, Ask Questions, or Get More Information about How to Use the SAS Clinical Standards Toolkit?

See “References” in Chapter 1, “Introduction to the SAS Clinical Standards Toolkit,” in the *SAS Clinical Standards Toolkit: User's Guide*.

Can the SAS Clinical Standards Toolkit 1.5 Be Used with Any Release of SAS Besides SAS 9.3?

The SAS Clinical Standards Toolkit 1.5 release is validated only on SAS 9.3, running on these platforms:

- Windows 32
- Windows for x64
- Linux for x64
- Solaris SPARC

In general, the code base for the SAS Clinical Standards Toolkit is designed to be backwardly compatible with prior releases of Base SAS, with only limited use of new features introduced with recent SAS releases. The open-source nature of the product allows user updates to all product components. “Off-label” use of the product is at the discretion of the user.

How Is Availability of a New SAS Clinical Standards Toolkit Release Publicized?

Releases of the SAS Clinical Standards Toolkit occur periodically. Please have your site representative contact your unit's SAS account manager for updates on the release schedule.

Availability of product updates is typically announced on the SAS and Clinical Trials Community forum (`http://communities.sas.com/community/sas_and_clinical_trials`). You can use e-mail notifications and RSS feeds to ensure that you do not miss any announcements.

The SAS Clinical Standards Toolkit development team members also report on product updates and availability at user conferences for specific industries such as PharmaSUG and PhUSE.

Does a New Version of the SAS Clinical Standards Toolkit Automatically Overwrite Files Installed with a Previous Version?

No. SAS has a strategy of adding the release number to folder names for both the sample study library files and the global standards library files. For example:

- SAS Clinical Standards Toolkit 1.4

  ***global standards library directory*/standards/cdisc-sdtm-3.1.2-1.4/control**

- SAS Clinical Standards Toolkit 1.5

  ***global standards library directory*/standards/cdisc-sdtm-3.1.2-1.5/control**

In addition, SAS does not automatically reset the default version of any previously installed version of a standard to the current version. Instead, you must make the decision to upgrade to the current version.

See “Unregistering an Old Version of a Standard, and Then Registering a New Version of a Standard” in Chapter 2, “Framework,” in the *SAS Clinical Standards Toolkit: User's Guide*.