SAS® Clinical Standards Toolkit 1.7: Operational Qualification, Second Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® Clinical Standards Toolkit 1.7: Operational Qualification, Second Edition*. Cary, NC: SAS Institute Inc.

SAS® Clinical Standards Toolkit 1.7: Operational Qualification, Second Edition

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2016

1.7-P1:clinstdtktiq

Contents

PART 5 Define-XML 57

PART 6 Dataset-XML 75

Part 1

Before You Begin

1

Introduction

Purpose

Starting with SAS Clinical Standards Toolkit 1.5, an internal validation process is provided. This process has been designed using tools and metadata already available with the product. In other words, the SAS Clinical Standards Toolkit is set up as a standard within the product and is validated against a set of reference metadata.

The SAS Clinical Standards Toolkit internal validation addresses two primary use cases:

- Perform installation qualification and operational qualification.

  This is implemented with and illustrated by the use of the validate_iqoq sample driver, located in the ***`sample study library directory`***`/cst-framework-1.7/programs` folder. This is a two-step process.

1. Select the CST-FRAMEWORK standard, and run the checks that are defined in the validation_control_glmeta view of the internal validation validation_master data set. This is a set of 64 checks (checkid < CSTV100) that look only at the global standards library metadata folder.
2. Select 1 to *n* specific standards, and run the checks that are defined in the validation_control_stdiqoq view of the internal validation validation_master data set. This is a set of 50 checks (checkid > CSTV100 that are relevant to installation qualification and operational qualification issues) that look only at metadata libraries other than the global standards library metadata folder.

- Perform validation on standard-specific metadata.

  This is implemented with and illustrated by the use of the validate_standard sample driver. Select 1 to *n* specific standards, and run the checks that are defined in the validation_control_std view of the internal validation validation_master data set.

  This is a set of 73 checks (checkid > CSTV100) that look only at metadata libraries other than the global standards library metadata folder.

For the purpose of operational qualification, this document implements the SAS Clinical Standards Toolkit internal validation for installation qualification and operational qualification.

For more information about internal validation, see Chapter 8, “Internal Validation,” in the *SAS Clinical Standards Toolkit: User's Guide*.

This document explains how to verify that the SAS Clinical Standards Toolkit 1.7 has been installed correctly and is operating properly. The installation is tested by running a series of SAS Clinical Standards Toolkit internal validation programs. These programs must be run in the sequence that they are presented. In addition to the internal validation process, a separate process enables you to manually run driver programs to verify that the product is operating correctly.

Note: Driver programs for the standards (for example, ODM, CRT-DDS, and SDTM) that are supplied with the SAS Clinical Standards Toolkit run off of the supplied sample data. The sample data is not clean. Therefore, errors and warnings will be present in the resulting data sets. This is normal.

Assumptions and Notes

General Assumptions

- The second maintenance release for SAS 9.4 has been installed and is functioning correctly.
- The SAS Clinical Standards Toolkit 1.7 has been installed, including the sample study libraries.
- The person running these tests is familiar with running SAS programs. This includes being able to submit SAS programs via the Program Editor, review the SAS log, and review the contents of SAS data sets.
- The installation of the SAS Clinical Standards Toolkit has not been modified from the default installation. If the sample studies have been modified before running these tests, your results data sets can vary from what is described in this document.

  Note: With a default installation, the results data sets must not contain errors or warnings. With a modified installation, errors or warnings might be normal, but they must be resolved by you.

File Path Separator

This document is used for both UNIX and Microsoft Windows environments. The forward slash character (/) is used in file paths as the separator between path components, which works in both operating system environments.

sample study library directory within This Document

sample study library directory is used to denote the sample study libraries available with SAS Clinical Standards Toolkit 1.7.

The default value for SAS Clinical Standards Toolkit 1.7 on Microsoft Windows is **`C:/cstSampleLibrary`**.

Variables Referred to by the Tests

The tests refer to these variables, which are defined relative to *sample study library directory*. When running the tests, substitute the variables with these associated paths:

- *CST_FRAMEWORK*

  ***sample study library directory*/cst-framework-1.7**

- *CST_SDTM*

  ***sample study library directory*/cdisc-sdtm-3.1.3-1.7/sascstdemodata**

- *CST_ODM*

  ***sample study library directory*/cdisc-odm-1.3.1-1.7**

- *CST_CRTDDS*

  ***sample study library directory*/cdisc-crtdds-1.0-1.7**

- *CST_DEFINEXML*

  ***sample study library directory*/cdisc-definexml-2.0.0-1.7**

- CST_DATASETXML

  ***sample study library directory*/cdisc-datasetxml-1.0.0-1.7**

Generation of a PDF File

The last manual test (see Chapter 21, “Test 5: Report Check Metadata,” on page 105) generates a PDF file. On Microsoft Windows, when a PDF file is generated, the PDF should automatically appear in a browser window. On UNIX, if you have not set up the SAS configuration variable SAS.helpBrowser, you see this message:

```
The requested information could not be displayed because the
connection to the remote browser server failed.
```

Click **OK** to continue.

Configure your UNIX SAS environment to support a browser that can display PDF files. Or, copy the PDF file to an environment where you can display it.

The Standards in This Document

The parts in this document that describe the standards are samples of several standards from the SAS Clinical Standards Toolkit. Each part describes how to access the sample study data using the driver programs to verify that the data, the metadata, and the SAS Clinical Standards Toolkit macros are functioning properly.

Part 2

Internal Validation

2

Installation Qualification and Operational Qualification

Introduction

Installation qualification and operational qualification is implemented with and illustrated by the use of the validate_iqoq sample driver, located in the ***sample study library directory*/cst-framework-1.7/programs** folder. This is a two-step process.

1. Select the CST-FRAMEWORK standard, and run the checks that are defined in the validation_control_glmeta view of the internal validation validation_master data set.

   This is a set of 64 checks (checkid < CSTV100) that look only at the global standards library metadata folder.

2. Select 1 to *n* specific standards, and run the checks that are defined in the validation_control_stdiqoq view of the internal validation validation_master data set.

   This is a set of 50 checks (checkid > CSTV100 that are relevant to installation qualification and operational qualification issues) that look only at the metadata libraries other than the global standards library metadata folder.

Note: The validation Results data set that is generated by the internal validation installation qualification and operational qualification contains many observations. Your number of observations can differ from the numbers shown in this document due to installation configurations that differ from a default installation of the SAS Clinical Standards Toolkit. For example, CDISC SEND might not be installed.

Steps

1. Start a new SAS session.

2. In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_FRAMEWORK*/programs/validate_iqoq.sas**.

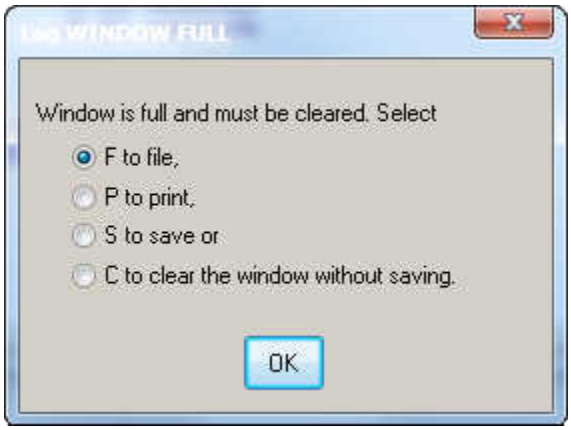
3. Select **Run** ▶ **Submit**.

   The program writes to the SAS log file and creates a cstrslt.validation_results data set in the ***CST_FRAMEWORK*/results** directory.

   Note: The SAS log might reach its limit depending on your system options. If it does, print the contents to a file, and select **APPEND** in the pop-up menu. This can happen several times during the run, so be sure to append each time it happens. To maximize the log size, you can add the option `-DMSLOGSIZE 999999` to the SAS configuration file.

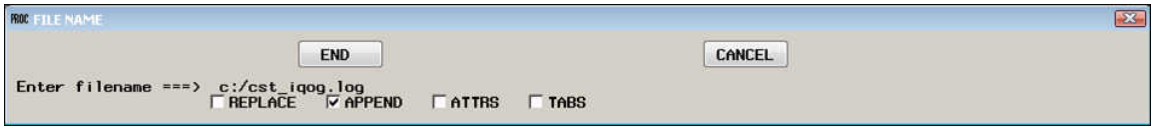
4. If the SAS log reaches it limit, perform these steps:

   a. In the pop-up window, select **F to file**.



b Enter a filename, and select **APPEND** or **REPLACE**.

Note: Select **REPLACE** for the first occurrence of the pop-up window only.



c Click **END**.

d Repeat steps a through c until finished.

Use the same filename each time, and select **APPEND**.

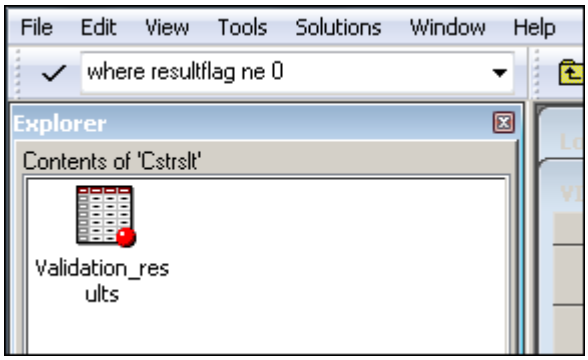
5 Review the log to ensure that there are no errors or warnings.

6 The column labeled **Process status** (named _cst_rc in the cstrslt.validation_results data set) is **0** for all records.

7 Review the cstrslt.validation_results data set using the SAS Explorer, especially for these conditions:

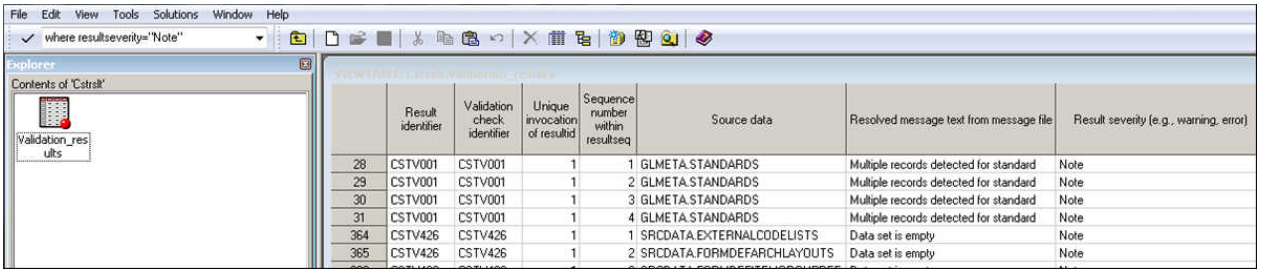
a The field **resultflag** is not 0.

When this value is not 0, a potential problem might exist. To more easily check this value, subset the validation_results data set by entering `where resultflag ne 0` in the control box in the upper left:



b A number of observations can have **result flag=1** or **result flag=-1**.

If the **Result severity** column is **Note**, these values are acceptable. To more easily check these values, subset the validation_results data set by entering `where resultseverity = "Note"` (this is case sensitive) in the control box in the upper left:



	Result identifier	Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)
28	CSTV001	CSTV001	1	1	GLMETA.STANDARDS	Multiple records detected for standard	Note
29	CSTV001	CSTV001	1	2	GLMETA.STANDARDS	Multiple records detected for standard	Note
30	CSTV001	CSTV001	1	3	GLMETA.STANDARDS	Multiple records detected for standard	Note
31	CSTV001	CSTV001	1	4	GLMETA.STANDARDS	Multiple records detected for standard	Note
364	CSTV426	CSTV426	1	1	SRCDATA.EXTERNALCODELISTS	Data set is empty	Note
365	CSTV426	CSTV426	1	2	SRCDATA.FORMDEFARCHLAYOUTS	Data set is empty	Note

Here are examples of where resultseverity="Note" or "Info" and resultflag ne 0:

- In this example of where resultseverity="Note", multiple records are detected because there are multiple standard versions for ODM (1.3.0 and 1.3.1) and SDTM (3.1.2, 3.1.3, and 3.2). If multiple records were found for the same standard version, this check would be in error.

Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)	Actual value observed	Record-level keys + values
CSTV001	1	1	GLMETA.STANDARDS	Multiple records detected for standard	Note	1	0	keys=standard mnemonic	standard=CDISC-ODM,mnemonic=ODM
CSTV001	1	2	GLMETA.STANDARDS	Multiple records detected for standard	Note	1	0	keys=standard mnemonic	standard=CDISC-SDTM,mnemonic=SDTM
CSTV001	1	3	GLMETA.STANDARDS	Multiple records detected for standard	Note	1	0	keys=standard mnemonic	standard=CDISC-SDTM,mnemonic=SDTM

- In this example of where resultseverity="Info", a check was not run because the check is not applicable to this standard. An informational check informs you that check CSTV251 is not applicable to this standard.

Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)
CSTV251	2	1	[CSTMETA.STANDARDS][	Check not run, not applicable to this standard	Info	-1	0
CSTV252	3	1	[CSTMETA.STANDARDSA	Check not run, not applicable to this standard	Info	-1	0
CSTV252	4	1	[CSTMETA.STANDARDSA	Check not run, not applicable to this standard	Info	-1	0

- In this example of where resultseverity="Info", a check was not run because check CSTV262, included with the SAS Clinical Standards Toolkit, has not yet been implemented in this release. Therefore, the check did not run.

Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)	Actual value observed	Basis or explanation for result
CSTV262	2	1	CSTVALIDATE	Check not run, checkstatus < 1	Info	-1	0	checkstatus=-2 (not implemented in this release)	Excludes checkstatus=0 (inactive), -1 (deprecated/archived), -2 (not implemented in this release)

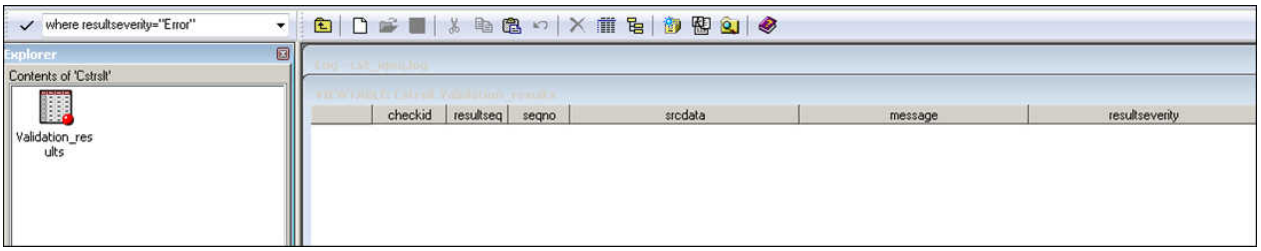
- In this example of where resultseverity="Note", these data sets are empty. They are empty because they are templates and do not contain observations.

Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)	Actual value observed
CSTV426	1	1	SRCDATA.EXTERNALCODELISTS	Data set is empty	Note	1	0	
CSTV426	1	2	SRCDATA.FORMDEFARCHLAYOUTS	Data set is empty	Note	1	0	
CSTV426	1	3	SRCDATA.FORMDEFITEMGROUPREFS	Data set is empty	Note	1	0	
CSTV426	1	4	SRCDATA.FORMDEFS	Data set is empty	Note	1	0	

- In this example of where resultseverity="Info", **Result severity** equals **Info** because the controlled terminology is not associated with an sl_cntl folder. There are no control type data sets associated with controlled terminology.

Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)	Actual value observed
	1	3	_CSTREADSTDS	SAS DATASET (sl_cntl.stdvalidation_sasrefs) does not exist	Info	1	0	

No observations should appear when you enter `where resultseverity = "Error"` in the control box in the upper left:



Any observation meeting the criterion `where resultseverity = "Warning"` must be assessed individually. For example, in the validation of the CDISC Define-XML 2.0.0 standard, this result might be reported:

	resultid	checkid	resultseq	seqno	srcdata	message	resultseverity	resultflag	_cst_rc
88	CSTV275	CSTV275	3	14	REFCNTL.VALIDATION_MASTER	Data set is empty	Warning	1	0

This message indicates that internal validation is correctly reporting that the validation_master data set for CDISC Define-XML 2.0.0 is empty because validation of the SAS representation of CDISC Define-XML 2.0.0 was not implemented in the SAS Clinical Standards Toolkit.

Another example in which observations meet the criterion `where resultseverity = "Warning"` is this result:

Result identifier	Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)
CSTV275	CSTV275	1	5	REFMETA.REFERENCE_ITEMGROUPS	Data set is empty	Warning	1	0
CSTV275	CSTV275	1	7	REFMETA.REFERENCE_VALUES	Data set is empty	Warning	1	0
CSTV275	CSTV275	3	19	REFMETA.REFERENCE_ITEMGROUPS	Data set is empty	Warning	1	0
CSTV275	CSTV275	3	21	REFMETA.REFERENCE_VALUES	Data set is empty	Warning	1	0

This message indicates that the refererence_values and reference_itemgroups data sets are empty.

Note: The CDASH 1.1 sample data sets refererence_values and reference_itemgroups supplied by SAS are intentionally empty. These data sets are specific to each customer. SAS cannot anticipate the CDASH representation that would populate the metadata in these data sets. SAS expects that this metadata is defined during implementation of the SAS Clinical Standards Toolkit. A warning is appropriate as a part of the internal validation of the CDASH standard.

8 Close the SAS session.

Running the validate_iqoq internal validation program without error confirms that all metadata is in place, all files are in place, and all access (whether Read or Write) to the SAS Clinical Standards Toolkit is properly initialized. This process ensures that the installation of the SAS Clinical Standards Toolkit was done properly and that the key components are operational.

Sign-Off

Internal Validation - Installation Qualification and Operational Qualification
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

Part 3

ODM

3

Test 1: Create SAS ODM from XML

Introduction

This test reads a CDISC ODM 1.3.1 XML file and builds a SAS representation of the metadata that is defined in the XML.

Note: To run against ODM 1.3.0, use the same information in this section, but substitute 1.3.1 with 1.3.0. Running against ODM 1.3.0 creates fewer data sets and less content (rows) within data sets.

Steps

1. Start a new SAS session.
2. In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_ODM*/programs/create_sasodm_fromxml.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates data sets in the formats, metadata, and data subdirectories in the ***CST_ODM*/derived** directory. It creates a read_results data set in the ***CST_ODM*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the read_results data set in the ***CST_ODM*/results** directory to ensure that these conditions are met:

TIP In the SAS Explorer, you can view it as **Read_results** in the **Results** library.

- The message column contains correct paths and process metadata.
- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- A record reports that the ODM file was read successfully.

6 Review the ***CST_ODM*/derived/metadata** directory to ensure that these conditions are met:

TIP In the SAS Explorer, you can view these data sets in the **Srcmeta** library.

- The directory contains two data sets: source_tables and source_columns.
- The source_tables data set contains 76 rows and 10 columns.
- The source_columns data set contains 352 rows and 16 columns.

7 Review the ***CST_ODM*/derived/data** directory to ensure that these conditions are met:

- There are 76 new SAS data sets. (Do not count any data sets that are not SAS, such as .xpt files.)
- The codelists data set contains 23 records and five columns.

8 Close the SAS session.

Sign-Off

Test 1: Create SAS ODM from XML
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

4

Test 2: Validate SAS ODM

Introduction

This test validates a SAS representation of the metadata that is defined in the CDISC ODM 1.3.1 XML file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_ODM*/programs/validate_odm_data.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates a validation_results data set and a validation_metrics data set in the ***CST_ODM*/results** directory.

Note: This program can fill up the log window if running interactively. If so, save the output of the log to a file when prompted to do so.

4 Review the log to ensure that there are no errors or warnings.

5 Review the validation_results data set in the ***CST_ODM*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **validation_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records. There are two failures of ODM0110.
- The data set contains 385 records.
- There are two records with **resultflag=1** and **resultseverity="Error"**. Both of these records have **checkid="ODM0110"**.

  Note: The errors messages are expected and are included in the sample data to cause a validation error for demonstration purposes.

219	ODM0110	34	1	SRCDATA.ITEMDEFS (SRCDATA.CODELISTS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDEFS	Error	1	0	CODELISTREF=CodeLists.OID.LBTEST
295	ODM0110	110	1	SRCDATA.ANNOTATIONFLAG (SRCDATA.CODELISTS)	The foreign key OID does not have a corresponding value in the target data set SRCDATA.ANNOTATIONFLAG	Error	1	0	FLAGTYPECODELISTOID=CodeLists.OID.dmgmt.req_ig

6 Review the validation_metrics data set in the ***CST_ODM*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **validation_metrics** in the **Results** library.

- The data set contains 656 records.
- The last record reports that there were two records with **"Content errors, warnings and notes"**.

7 Close the SAS session.

Sign-Off

Test 2: Validate SAS ODM
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

5

Test 3: Create ODM XML from SAS ODM

Introduction

This test creates a CDISC ODM 1.3.1 XML file from the SAS representation of the metadata.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_ODM*/programs/create_odmxml.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates an XML file, odm_sample_out.xml, in the ***CST_ODM*/sourcexml** directory. It creates a write_results data set in the ***CST_ODM*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the write_results data set in the ***CST_ODM*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **write_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for all records.
- The data set contains 70 records.
- There is a record where **Source data** is **ODM_WRITE** that reports that the ODM file was created.

6 Ensure that the **CST_ODM/sourcexml** directory contains a new XML file odm_sample_out.xml.

If you were to compare the file odm_sample_out.xml to the file odm_sample.xml in the same directory, you would see that the only difference is the ODM/@CreationDateTime attribute.

7 Close the SAS session.

Sign-Off

Test 3: Create ODM XML from SAS ODM
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

6

Test 4: Find Unsupported Tags in ODM XML

Introduction

This test parses a CDISC ODM 1.3.1 XML file and finds elements and attributes that the SAS Clinical Standards Toolkit does not recognize by default. These elements and attributes might be vendor extensions, customer extensions, or new tags implemented in a later version of ODM.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_ODM*/programs/find_unsupported_tags.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates a readxmltags_results data set in the ***CST_ODM*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the readxmltags_results data set in the ***CST_ODM*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **readxmltags_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for eight records, and **1** for all other records.
- The data set contains 28 records.
- There are four records that contain **checkid="ODM0900"** and 16 records that contain **checkid="ODM0901"**. For the ODM0900 check, the message indicates **"Element found in XML file that is not present in CDISC ODM Model"**. For the ODM0901 check, the message indicates **"Attribute found in XML file that is not present in CDISC ODM Model"**.

6 Close the SAS session.

Sign-Off

Test 4: Find Unsupported Tags in ODM XML
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

Part 4

CRT-DDS

7

Test 1: Validate CRT-DDS

Introduction

This test validates a SAS representation of the metadata and data that is defined in the SAS representation of the CRT-DDS 1.0 model.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_CRTDDS*/programs/validate_crtdds_data.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates a validation_results data set and a validation_metrics data set in the ***CST_CRTDDS*/results** directory.

Note: This program can fill up the log window if running interactively. If so, save the output of the log to a file when prompted to do so.

4 Review the log to ensure that there are no errors or warnings.

5 Review the validation_results data set in the ***CST_CRTDDS*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **validation_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The data set contains 202 records.
- There are 21 records that contain **"Warning: Check not run"**. These records contain **checkid="CRT0100"** and **resultid="CST0022"**.

  These warnings are the result of missing information, such as key variables. Because these warnings apply to the metadata, a warning is issued, and the check does not run.

6 Review the validation_metrics data set in the ***CST_CRTDDS*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **validation_metrics** in the **Results** library.

- The data set contains 326 records.
- The last record reports that there were 21 records with **"Content errors, warnings and notes"**.

7 Close the SAS session.

Sign-Off

Test 1: Validate CRT-DDS
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

8

Test 2: Create SAS CRT-DDS from SDTM Metadata

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit derived CRT-DDS 1.0 metadata from an SDTM study as a prerequisite to building a define.xml file in Test 5.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select **CST_CRTDDS/programs/create_crtdds_from_sdtm.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates 39 data sets in the **`CST_CRTDDS/data`** directory. It create a Results data set in the **`CST_CRTDDS/results`** directory.

4 Review the log to see whether there are any errors or warnings.

5 Review the ***`CST_CRTDDS`***`/data` directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it in the **srcdata** library.

- There are 39 new SAS data sets.
- The codelists data set has 46 records and 5 columns.

6 Review the sdtmtodefine_results data set in the ***`CST_CRTDDS`***`/results` directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **sdtmtodefine_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The data set contains 40 records.

7 Close the SAS session.

Sign-Off

Test 2: Create SAS CRT-DDS from SDTM Metadata
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

9

Test 3: Create SAS CRT-DDS from define.xml

Introduction

This test creates a CRT-DDS 1.0 SAS representation file from a define.xml file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_CRTDDS*/programs/create_sascrtdds_fromxml.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates the SAS representation of the CRT-DDS data sets in the ***CST_CRTDDS*/derived data** directory.

TIP In the SAS Explorer, you can view it in the **srcdata** library.

4 Review the log to ensure that there are no errors or warnings.

5 Review the read_results data set in the ***CST_CRTDDS*/results** directory to ensure that these conditions are met:

TIP In the SAS Explorer, you can view it as **read_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for all records.
- There is a record where **Source data** is **CRTDDS_READ** that reports that the define.xml file was read successfully.
- There is a record where **Source data** is **JAVA CHECK** that reports **No java issues**.

6 Ensure that the ***CST_CRTDDS*/derived data** directory contains 39 SAS data sets that represent the SAS interpretation of the CRT-DDS format.

7 Open the clitemdecodetranslatedtext SAS data set.

It contains 4838 observations. The first 17 observations are shown in this figure.

	Human-readable text appropriate for a particular language	Natural language or country-specific language variant	Foreign key: CodeListItems.OID
1	DOSE INCREASED	en	N77970
2	DOSE NOT CHANGED	en	N77981
3	DOSE REDUCED	en	N77992
4	DRUG INTERRUPTED	en	N78003
5	DRUG WITHDRAWN	en	N78014
6	NOT APPLICABLE	en	N78025
7	UNKNOWN	en	N78036
8	MILD	en	N78054
9	MODERATE	en	N78065
10	SEVERE	en	N78076
11	DAYS	en	N78094
12	HOURS	en	N78105
13	MONTHS	en	N78116
14	WEEKS	en	N78127
15	YEARS	en	N78138
16	GBR	en	N78156
17	USA	en	N78167

8 Close the SAS session.

Sign-Off

Test 3: Create SAS CRT-DDS from define.xml
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

10

Test 4: Create define.xml

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit Java and XML-related libraries are installed correctly. The SAS Clinical Standards Toolkit and libraries can create a CRT-DDS file (define.xml).

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select **CST_CRTDDS/programs/create_crtdds_define.sas**.

   This program writes to the SAS log and generates two files in the ***CST_CRTDDS/*****sourcexml** directory. It creates a Results data set in the ***CST_CRTDDS/*****results** directory.

3 Select **Run** ▶ **Submit**.

4 Ensure that two files were generated in the ***CST_CRTDDS*/sourcexml** directory: define.xml and define-v1-updated-html.xsl.

5 Open the define.xml file.

On Microsoft Windows, you can open it by double-clicking it in the SAS Program Editor. This renders the file in your default web browser or any other application that has been associated with XML files.

On UNIX, if you have not set up your browser configuration in SAS, you need to copy define.xml and define-v1-updated-html.xsl to an environment where you can display the define.xml file in a web browser.

Note: The style sheet information in define-v1-updated-html.xsl is not guaranteed to work for all browser types and versions to produce the correct HTML, but it does work for Internet Explorer 6.0 and higher.

6 Ensure that the first few rows of the first table appear similar to this image:

SDTM Datasets for Study study1

Dataset	Description	Class	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
CO	Comments	Special Purpose Domains	One record per comment per subject	Tabulation	STUDYID, USUBJID, COSEQ	Comments SAS transport file
DA	Drug Accountability	Findings	One record per drug accountability finding per subject	Tabulation	STUDYID, USUBJID, DATESTCD, DADTC	Drug Accountability SAS transport file
DM	Demographics	Special Purpose Domains	One record per subject	Tabulation	STUDYID, USUBJID	Demographics SAS transport file
DS	Disposition	Events	One record per disposition status or protocol milestone per subject	Tabulation	STUDYID, USUBJID, DSDECOD, DSSTDTC	Disposition SAS transport file
DV	Protocol Deviations	Events	One record per protocol deviation per subject	Tabulation	STUDYID, USUBJID, DVTERM, DVSTDTC	Protocol Deviations SAS transport file
EG	ECG Test Results	Findings	One record per ECG observation per time point per visit per subject	Tabulation	STUDYID, USUBJID, EGTESTCD, VISITNUM, EGTPTREF, EGTPTNUM	ECG Test Results SAS transport file
EX	Exposure	Interventions	One record per constant dosing interval per subject	Tabulation	STUDYID, USUBJID, EXTRT, EXSTDTC	Exposure SAS transport file

Note: Minor variations in appearance are possible and are not a problem. Reviewing these rows is sufficient to confirm that the product is installed and functioning properly.

7 Ensure that the define.xml file contains tables for the following:

- For each domain, a table that lists the domain's variables

- Computational Algorithms Section
- Controlled Terminology

8 In the Controlled Terminology section of the define.xml file, ensure that the Code List is VSTESTCD, including the values BMI and WEIGHT.

Code List - VSTESTCD, Reference Name (CL.VSTESTCD)

Coded Value	Decode
ABSKNF	ABSKNF
BMI	BMI
BODLNGTH	BODLNGTH
BODYFAT	BODYFAT
BSA	BSA
DIABP	DIABP
FARMCIR	FARMCIR
FRMSIZE	FRMSIZE
HDCIRC	HDCIRC
HEIGHT	HEIGHT
HIPCIR	HIPCIR
HR	HR
KNEEHEEL	KNEEHEEL
LBM	LBM
MAP	MAP
OXYSAT	OXYSAT
PULSE	PULSE
PULSEPR	PULSEPR
RESP	RESP
SAD	SAD
SSSKNF	SSSKNF
SYSBP	SYSBP
TBW	TBW
TEMP	TEMP
TRSKNF	TRSKNF
WEIGHT	WEIGHT
WSTCIR	WSTCIR

9 Close the SAS session.

Sign-Off

Test 5: Create define.xml
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

Part 5

Define-XML

11

Test 1: Create Define-XML 2.0 SAS Data Sets from SDTM Source Metadata

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit derived Define-XML 2.0 metadata from an SDTM study as a prerequisite to building a define.xml file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_DEFINEXML*/programs/create_sasdefine_from_source.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates 31 data sets in the ***CST_DEFINEXML*/data/cdisc-sdtm-3.1.2** directory. It creates a Results data set in the ***CST_DEFINEXML*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the ***CST_DEFINEXML*/data** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it in the **srcdata** library.

- There are 31 new SAS data sets.
- The itemdefs data set contains 535 records and 13 columns.

6 Review the sourcetodefine_results data set in the ***CST_DEFINEXML*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **sourcetodefine_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records
- The data set contains 74 records.

7 Close the SAS session.

Sign-Off

Test 1: Create Define-XML 2.0 SAS Data Sets from SDTM Source Metadata
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

12

Test 2: Create Define-XML 2.0 File from SAS Data Sets

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit libraries that are related to Java and XML are installed correctly. The SAS Clinical Standards Toolkit can create a Define-XML 2.0 file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_DEFINEXML*/programs/create_definexml.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates two files in the ***CST_DEFINEXML*/sourcexml** directory. It creates a Results data set in the ***CST_DEFINEXML*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Ensure that two files were created in the ***CST_DEFINEXML*/sourcexml** directory: define-sdtm-3.1.2.xml and define2-0-0.xsl.

6 Open the define-sdtm-3.1.2.xml file.

On Microsoft Windows, you can open it by double-clicking it in the SAS Program Editor. This renders the file in your default web browser or any other application that has been associated with XML files.

On UNIX, if you have not set up your browser configuration in SAS, you need to copy define-sdtm-3.1.2.xml and define2-0-0.xsl to an environment where you can display the XML file in a web browser.

Note: The style sheet information in define2-0-0.xsl is not guaranteed to work for all browser types and versions to produce the correct HTML, but it does work for Internet Explorer 6.0 and higher. The Chrome browser, for example, does not allow local XML and XSLT processing. Depending on your browser, you might see a security warning because the style sheet uses Javascript.

7 Ensure that the display looks similar to this image:

ADaM-IG 1.0

Date of Define-XML document generation: 2016-02-16T15:11:22-05:00

Stylesheet version: 2016-02-11

- Analysis Data Reviewer's Guide
- Clinical Study Report
- Statistical Analysis Plan
- ▸ Analysis Results Metadata
- ▸ Analysis Datasets
- ▸ Parameter Value Level Metadata
- ▸ Controlled Terminology
- ▸ Analysis Derivations
- ▸ Comments

Standard	ADaM-IG 1.0
Study Name	CDISC-Sample
Study Description	CDISC-Sample Data Definition
Protocol Name	CDISC-Sample
Metadata Name	Data Definitions for CDISC-Sample, ADaM-IG 1.0
Metadata Description	Data Definitions for CDISC-Sample, ADaM-IG 1.0

Analysis Results Metadata (Summary) for Study CDISC-Sample

Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)
- Dose response analysis for ADAS-Cog changes from baseline
- Pairwise comparisons to placebo for ADAS-Cog changes from baseline

Table 14-5.02 Incidence of Treatment Emergent Serious Adverse Events by Treatment Group
- Incidence of Treatment Emergent Serious Adverse Events by Treatment Group

Analysis Results Metadata (Detail) for Study CDISC-Sample

Table 14-3.01

Display	**Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)**
Analysis Result	**Dose response analysis for ADAS-Cog changes from baseline**
Analysis Parameter(s)	PARAMCD = "ACTOT" (Adas-Cog(11) Subscore)

Note: Minor variations in appearance are possible and are not a problem. Reviewing the display is sufficient to confirm that the product is installed and functioning properly.

8 Ensure that the last few rows (indicating that comments are being displayed) appear similar to this image:

COM.VS.VSSTRESU	Standard units consistent with CDISC controlled terminology
COM.SUPPQSCG.QVAL.WC.SUPPQSCG.QVAL.00087	QSMM-CRF Page 13; QSCS-CRF Pages 14; QSCG-CRF Page 17
COM.SUPPQSCS.QVAL.WC.SUPPQSCS.QVAL.00088	QSMM-CRF Page 13; QSCS-CRF Pages 14; QSCG-CRF Page 17
COM.SUPPQSMM.QVAL.WC.SUPPQSMM.QVAL.00089	QSMM-CRF Page 13; QSCS-CRF Pages 14; QSCG-CRF Page 17

Go to the top of the define.xml

9 Review the write_results data set in the ***CST_DEFINEXML/results*** directory to ensure that these conditions are met:

TIP In the SAS Explorer, you can view it as **write_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for all records.
- The data set contains 79 records.

- There is a record where **Source data** is **DEFINE_WRITE** that reports that the XML file was created.
- There is a record where **Source data** is **XML TRANSFORMER** that reports **The document validated successfully**.

10 Close the SAS session.

Sign-Off

Test 2: Create Define-XML 2.0 File from SAS Data Sets
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

13

Test 3: Create SAS Data Sets from Define-XML 2.0 File

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit derived a SAS representation of the metadata from a Define-XML 2.0 file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_DEFINEXML*/programs/create_sasdefine_fromxml.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file, creates a Define-XML 2.0 SAS representation in the ***CST_DEFINEXML*/derivedata/cdisc-sdtm-3.1.2** directory from the ***CST_DEFINEXML*/sourcexml/define2-0-0-example-sdtm.xml** file. It creates a Results data set in the ***CST_DEFINEXML*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the ***CST_DEFINEXML*/derivedata/cdisc-sdtm-3.1.2** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it in the **srcdata** library.

- There are 54 new SAS data sets that represent the SAS interpretation of the metadata in the define2-0-0-example-sdtm.xml file.
- The itemdefs data set contains 535 records and 13 columns.

6 Review the read_results data set in the ***CST_DEFINEXML*/results** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **read_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for all records.
- The data set contains 79 records.
- There is a record where **Source data** is **DEFINE_READ** that reports that the XML file was read successfully.
- There is a record where **Source data** is **XML TRANSFORMER** that reports **The document validated successfully**.

7 Close the SAS session.

Sign-Off

Test 3: Create SAS Data Sets from Define-XML 2.0 File
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

14

Test 4: Create Define-XML 2.0 File (Including Analysis Results Metadata) from SAS ADaM Data Sets

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit libraries related to Java and XML have been installed correctly. The SAS Clinical Standards Toolkit can create a Define-XML 2.0 file including Analysis Results Metadata.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_DEFINEXML*/programs/create_definexml_from_source_adam.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates two files in the ***CST_DEFINEXML*/sourcexml** directory. It creates a Results data set in the ***CST_DEFINEXML*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Ensure that two files were created in the ***CST_DEFINEXML*/sourcexml** directory: define-adam-2.1.xml and define2-0-0.xsl.

6 Open the define-adam-2.1.xml file.

On Microsoft Windows, you can open it by double-clicking it in the SAS Program Editor. This renders the file in your default web browser or any other application that has been associated with XML files.

On UNIX, if you have not set up your browser configuration in SAS, you need to copy define-adam-2.1.xml and define2-0-0.xsl to an environment where you can display the XML file in a web browser.

Note: The style sheet information in define2-0-0.xsl is not guaranteed to work for all browser types and versions to produce the correct HTML, but it does work for Internet Explorer 6.0 and higher. The Chrome browser, for example, does not allow local XML and XSLT processing. Depending on your browser, you might see a security warning because the style sheet uses Javascript.

7 Ensure that the display looks similar to this image:

ADaM-IG 1.0

Date of Define-XML document generation: 2016-02-16T15:11:22-05:00

Stylesheet version: 2016-02-11

- Analysis Data Reviewer's Guide
- Clinical Study Report
- Statistical Analysis Plan
- ▸ Analysis Results Metadata
- ▸ Analysis Datasets
- ▸ Parameter Value Level Metadata
- ▸ Controlled Terminology
- ▸ Analysis Derivations
- ▸ Comments

Standard	ADaM-IG 1.0
Study Name	CDISC-Sample
Study Description	CDISC-Sample Data Definition
Protocol Name	CDISC-Sample
Metadata Name	Data Definitions for CDISC-Sample, ADaM-IG 1.0
Metadata Description	Data Definitions for CDISC-Sample, ADaM-IG 1.0

Analysis Results Metadata (Summary) for Study CDISC-Sample

Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)
- Dose response analysis for ADAS-Cog changes from baseline
- Pairwise comparisons to placebo for ADAS-Cog changes from baseline

Table 14-5.02 Incidence of Treatment Emergent Serious Adverse Events by Treatment Group
- Incidence of Treatment Emergent Serious Adverse Events by Treatment Group

Analysis Results Metadata (Detail) for Study CDISC-Sample

Table 14-3.01

Display	**Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)**
Analysis Result	**Dose response analysis for ADAS-Cog changes from baseline**
Analysis Parameter(s)	PARAMCD = "ACTOT" (Adas-Cog(11) Subscore)

Note: Minor variations in appearance are possible and are not a problem. Reviewing the display is sufficient to confirm that the product is installed and functioning properly.

8 Review the sourcetodefinexml_adam_results data set in the ***CST_DEFINEXML/results*** directory to ensure that these conditions are met:

> **TIP** In the SAS Explorer, you can view it as **sourcetodefinexml_adam_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for all records.
- The data set contains 156 records.
- There is a record where **Source data** is **DEFINE_WRITE** that reports that the XML file was created.
- There is a record where **Source data** is **XML TRANSFORMER** that reports **The document validated successfully**.

9 Close the SAS session.

Sign-Off

Test 4: Create Define-XML 2.0 File (Including Analysis Results Metadata) from SAS ADaM Data Sets
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

Part 6

Dataset-XML

15

Test 1: Create Dataset-XML 1.0 Files from SDTM Source Data

Introduction

This test creates XML and ZIP files that confirm that the SAS Clinical Standards Toolkit creates Dataset-XML 1.0 files from an SDTM study as a prerequisite to building a define.xml file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_DATASETXML*/programs/create_datasetxml.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates 34 XML files and 34 ZIP files in the ***CST_DATASETXML*/sourcexml** directory. It creates a Results data set in the ***CST_DATASETXML*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the ***CST_DATASETXML*/sourcexml** directory to ensure that the following conditions are met:

- There are 34 new XML files.
- There are 34 new ZIP files.
- The ZIP file ae.zip contains one file (ae.xml).

6 Review the write_results data set in the ***CST_DATASETXML*/results** directory to ensure that the following conditions are met:

TIP In the SAS Explorer, you can view it as **write_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The data set contains 77 records.

7 Close the SAS session.

Sign-Off

Test 1: Create Dataset-XML 1.0 Files from SDTM Source Data
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

16

Test 2: Create SAS Data Sets from Dataset-XML 1.0 Files

Introduction

This test creates SAS data sets that confirm that the SAS Clinical Standards Toolkit derives SAS data sets from Dataset-XML 1.0 files.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_DATASETXML*/programs/create_sas_from_datasetxml.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates 34 new SAS data sets in the ***CST_DATASETXML*/data_derived** directory. It creates a Results data set in the ***CST_DATASETXML*/results** directory.

4 Review the log to ensure that there are no errors or warnings.

5 Review the ***CST_DATASETXML*/data_derived** directory to ensure that the following conditions are met:

> **TIP** In the SAS Explorer, you can view it in the **trgdata** library.

- There are 34 new SAS data sets.
- The AE data set contains 16 records and 18 columns.

6 Review the read_results data set in the ***CST_DATASETXML*/results** directory to ensure that the following conditions are met:

> **TIP** In the SAS Explorer, you can view it as **read_results** in the **Results** library.

- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The column named **resultflag** is **0** for all records.
- The data set contains 113 records.

7 Close the SAS session.

Sign-Off

Test 2: Create SAS Data Sets from Dataset-XML 1.0 Files
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

Part 7

SDTM

17

Test 1: Validate SDTM

Introduction

This test runs the sample program that is provided as part of the SDTM 3.1.3 standard. If this program runs successfully and produces the expected results, the SDTM 3.1.3 standard is correctly installed and functioning properly.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_SDTM*/programs/validate_data.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and generates a validation_results data set and a validation_metrics data set in the `CST_SDTM/data` directory.

TIP In the SAS Explorer, you can view these data sets in the **Results** library.

4 Review the log to ensure that there are no errors or warnings.

5 Review the validation_results data set in the `CST_SDTM/data` directory to ensure that these conditions are met:

- For the records where the column labeled **Validation check identifier** (named checkid) is blank, examine the message column to ensure that paths and process metadata are correct.
- The column labeled **Process status** (named _cst_rc) is **0** for all records.
- The data set contains 105 records.

6 Review the validation_metrics data set in the `CST_SDTM/data` directory and ensure that it contains these last few rows:

Metric Parameter	Count of Records
# of distinct check invocations	11
# check invocations not run	1
Errors (severity=High) reported	1
Warnings (severity=Medium) reported	3
Notes (severity=Low) reported	0
Structural errors, warnings and notes	0
Content errors, warnings and notes	5

7 Close the SAS session.

Sign-Off

Test 1: Validate SDTM
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

18

Test 2: Build Source Data

Introduction

This test references derived data from a CRT-DDS (define.xml) file to build a library of SDTM 3.1.3 domains.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_SDTM*/programs/create_sasdatafromxpt**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates an xpt_results data set in the ***CST_SDTM*/results** directory and 36 data sets in the ***CST_SDTM*/derived/data** directory.

Note: For this program, the library information was cleaned up, so these files are not immediately accessible under **Libraries** in the SAS Explorer. On Microsoft Windows, you can access these files through the SAS Explorer by navigating from within the SAS Explorer starting at the node labeled **My Computer**. On UNIX, it is necessary for you to copy these data sets into a directory that is viewable by the SAS Explorer (for example, your **Home Directory** listed under **Favorite Folders**).

4 Review the log to see whether there are any errors or warnings.

There should be no errors or warnings.

You might sporadically see warnings in the SAS log such as `WARNING: Libname <libref> is not assigned.` These occur with redundant requests to clear SAS librefs or filerefs and do not indicate a problem with the SAS Clinical Standards Toolkit installation.

5 Review the xpt_results data set to ensure that these conditions are met:

- The **Resolved message text from message file** column (named message) contains correct paths and process metadata.
- The column labeled **Process status** (named _cst_rc) is **0** for all records. There are not any checks with **resultseverity='Warning: Check not run'**.

	resultid	checkid	resultseq	seqno	srcdata	message	resultseverity	resultflag	_cst_rc
1	CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH C:\cstGlobalLibrary/standards/cst-frame	Info	0	0
2	CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to C:\cstGlobalLibrary/standards/cst-frame to perform the template lookup	Info	0	0
3	CST0102		1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences (SAS File and Library References) was created as requested	Info	0	0
4	CST0200		1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\gelgh\AppData\Local\Temp\ Temporary Files_TD9196_L73859_/sasreferences	Info	0	0
5	CST0200		1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info	0	0
6	CST0200		1	2	CSTUTIL_ALLOCATESASREFERENCE	SASReferences data set was successfully validated	Info	0	0
7	CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH C:\cstGlobalLibrary/standards/cdisc-sdt	Info	0	0
8	CST0200		1	1	SDTMUTIL_CREATESASDATAFROMX	PROCESS STANDARD: CDISC-SDTM	Info	0	0
9	CST0200		1	2	SDTMUTIL_CREATESASDATAFROMX	PROCESS STANDARDVERSION: 3.1.3	Info	0	0
10	CST0200		1	3	SDTMUTIL_CREATESASDATAFROMX	PROCESS DRIVER: CREATE_SASDATAFROMXPT	Info	0	0
11	CST0200		1	4	SDTMUTIL_CREATESASDATAFROMX	PROCESS DATE: 2014-11-24T11:17:38	Info	0	0
12	CST0200		1	5	SDTMUTIL_CREATESASDATAFROMX	PROCESS TYPE: DATA DERIVATION	Info	0	0
13	CST0200		1	6	SDTMUTIL_CREATESASDATAFROMX	PROCESS SASREFERENCES: work._cstsasrefs	Info	0	0
14	CST0200		1	7	SDTMUTIL_CREATESASDATAFROMX	PROCESS STUDYROOTPATH: C:\cstSampleLibrary/cdisc-crtdds-1.0-1.7	Info	0	0
15	CST0200		1	8	SDTMUTIL_CREATESASDATAFROMX	PROCESS GLOBALLIBRARY: C:\cstGlobalLibrary	Info	0	0
16	CST0200		1	9	SDTMUTIL_CREATESASDATAFROMX	PROCESS CSTVERSION: 1.7	Info	0	0
17	CST0200		1	10	SDTMUTIL_CREATESASDATAFROMX	Process completed successfully	Info	0	0
18	CST0102		1	1	CSTUTIL_SAVERESULTS	results.xpt_results was created as requested	Info	0	0

- The data set contains 18 records. One of the last records reports **Process completed successfully**.

  Note: Values that refer to temporary directories, files, or **PROCESS DATE:** vary.

6 Review the ***CST_SDTM*/derived/data** directory to ensure that these conditions are met:

- There are 36 new SAS data sets.
- The dm data set has 70 records and 28 columns.

7 Close the SAS session.

Sign-Off

Test 2: Build Source Data
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

19

Test 3: Build Source Metadata

Introduction

This test references derived data from a CRT-DDS (define.xml) file to build a set of SDTM 3.1.3 metadata in a structure required by the SAS Clinical Standards Toolkit.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_SDTM*/programs/create_sourcemetadata.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and creates data sets in the `CST_SDTM/results` directory and the `CST_SDTM/derived/metadata` directory.

Note: For this program, the library information was cleaned up, so these files are not immediately accessible under **Libraries** in the SAS Explorer. On Microsoft Windows, you can access these files through the SAS Explorer by navigating from within the SAS Explorer starting at the node labeled **My Computer**. On UNIX, it is necessary for you to copy these data sets into a directory that is viewable by the SAS Explorer (for example, your **Home Directory** listed under **Favorite Folders**).

4 Review the log to ensure that there are no errors or warnings.

5 Review the srcmeta_results data set in the ***CST_SDTM*/results** directory to ensure that these conditions are met:

- The **Resolved message text from message file** column (named message) contains correct paths and process metadata.
- The column labeled **Process status** (named _cst_rc) is **0** for all records.

	resultid	checkid	resultseq	seqno	srcdata	message	resultseverity	resultflag	_cst_rc
1	CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH C:\cstGlobalLibrary/standards/cst-frame	Info	0	0
2	CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to C:\cstGlobalLibrary/standards/cst-frame to perform the template lookup	Info	0	0
3	CST0102		1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences (SAS File and Library References) was created as requested	Info	0	0
4	CST0200		1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\geligh\AppData\Local\Temp\ Temporary Files_TD14680_L73859_/sasreference	Info	0	0
5	CST0200		1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info	0	0
6	CST0200		1	2	CSTUTIL_ALLOCATESASREFERENCE	SASReferences data set was successfully validated	Info	0	0
7	CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH C:\cstGlobalLibrary/standards/cdisc-sdt	Info	0	0
8	CST0200		1	1	SDTMUTIL_CREATESRCMETAFROMS	PROCESS STANDARD: CDISC-SDTM	Info	0	0
9	CST0200		1	2	SDTMUTIL_CREATESRCMETAFROMS	PROCESS STANDARDVERSION: 3.1.3	Info	0	0
10	CST0200		1	3	SDTMUTIL_CREATESRCMETAFROMS	PROCESS DRIVER: CREATE_SOURCEMETADATA	Info	0	0
11	CST0200		1	4	SDTMUTIL_CREATESRCMETAFROMS	PROCESS DATE: 2014-11-24T11:28:45	Info	0	0
12	CST0200		1	5	SDTMUTIL_CREATESRCMETAFROMS	PROCESS TYPE: METADATA DERIVATION	Info	0	0
13	CST0200		1	6	SDTMUTIL_CREATESRCMETAFROMS	PROCESS SASREFERENCES: work._cstsasrefs	Info	0	0
14	CST0200		1	7	SDTMUTIL_CREATESRCMETAFROMS	PROCESS STUDYROOTPATH: C:\cstSampleLibrary/cdisc-sdtm-3.1.3-1.7	Info	0	0
15	CST0200		1	8	SDTMUTIL_CREATESRCMETAFROMS	PROCESS GLOBALLIBRARY: C:\cstGlobalLibrary	Info	0	0
16	CST0200		1	9	SDTMUTIL_CREATESRCMETAFROMS	PROCESS CSTVERSION: 1.7	Info	0	0
17	CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref tmplt was allocated to C:\cstGlobalLibrary/standards/cdisc-sdt to perform the template lookup	Info	0	0
18	CST0102		1	2	CST_CREATEDSFROMTEMPLATE	trgmeta.source_study (Standard Study Metadata) was created as requested	Info	0	0
19	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7	Info	0	0
22	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7	Info	0	0
21	CST0102		1	2	CST_CREATEDSFROMTEMPLATE	trgmeta.source_values (Standard Value Metadata) was created as requested	Info	0	0
22	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7	Info	0	0
23	CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref tmplt was allocated to C:\cstGlobalLibrary/standards/cdisc-sdt to perform the template lookup	Info	0	0
24	CST0102		1	2	CST_CREATEDSFROMTEMPLATE	trgmeta.source_documents (Standard Document Metadata) was created as requested	Info	0	0
25	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7	Info	0	0
26	CST0074		1	4	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7	Info	0	0
27	CST0074		1	5	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7	Info	0	0
28	CST0102		1	1	CSTUTIL_SAVERESULTS	results.srcmeta_results was created as requested	Info	0	0

- The data set contains 28 records.

Note: Values that refer to temporary directories, files, or **PROCESS DATE:** vary.

- Where the **Result identifier** equals **CST0074**, the records report that study reference data was created in folder ***CST_SDTM*/derived/metadata**.

	resultid	checkid	resultseq	seqno	srcdata	message
19	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7\sascstdemodata\derived\metadata
22	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7\sascstdemodata\derived\metadata
25	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7\sascstdemodata\derived\metadata
26	CST0074		1	4	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7\sascstdemodata\derived\metadata
27	CST0074		1	5	SDTMUTIL_CREATESRCMETAFROMS	Study reference data created in C:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.7\sascstdemodata\derived\metadata

6 Review the ***CST_SDTM*/derived/metadata** directory to ensure that these conditions are met:

- There are five new data sets: source_columns, source_study, source_documents, source_values, and source_tables.
- The source_tables data set has 36 records and 15 columns.

7 Close the SAS session.

Sign-Off

Test 3: Build Source Metadata
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

20

Test 4: Build SAS Formats

Introduction

This test references derived data from a CRT-DDS (define.xml) file to build a SAS format catalog representing the codelists in the CRT-DDS (define.xml) file.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_SDTM*/programs/create_formatsfromcrtdds.sas**.

3 Select **Run** ▶ **Submit**.

The program writes to the SAS log file and creates a codelist_results data set in the ***CST_SDTM*/results** directory and creates a catalog named cterms in the ***CST_SDTM*/derived/formats** directory.

Note: For this program, the library information was cleaned up, so these files are not immediately accessible under **Libraries** in the SAS Explorer. On Microsoft Windows, you can access these files through the SAS Explorer by navigating from within the SAS Explorer starting at the node labeled **My Computer**. On UNIX, it is necessary for you to copy these data sets into a directory that is viewable by the SAS Explorer (for example, your **Home Directory** listed under **Favorite Folders**).

4 Review the log to ensure that there are no errors or warnings.

5 Review the codelist_results data set in the `CST_SDTM/results` directory to ensure that these conditions are met:

- The **Resolved message text from message file** column (named message) contains correct paths and process metadata.
- The column labeled **Process status** (named _cst_rc) is **0** for all records. There are not any checks with **resultseverity='Warning: Check not run'**.

	resultid	checkid	resultseq	seqno	srcdata	message	resultseverity	resultflag	_cst_rc
1	CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH C:\cstGlobalLibrary/standards/cst-frame	Info	0	0
2	CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to C:\cstGlobalLibrary/standards/cst-frame to perform the template lookup	Info	0	0
3	CST0102		1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences (SAS File and Library References) was created as requested	Info	0	0
4	CST0200		1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\geligh\AppData\Local\Temp\ Temporary Files_TD12004_L73859_/sasreference	Info	0	0
5	CST0200		1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info	0	0
6	CST0200		1	2	CSTUTIL_ALLOCATESASREFERENCES	SASReferences data set was successfully validated	Info	0	0
7	CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH C:\cstGlobalLibrary/standards/cdisc-sdt	Info	0	0
8	CST0200		1	1	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS STANDARD: CDISC-SDTM	Info	0	0
9	CST0200		1	2	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS STANDARDVERSION: 3.1.3	Info	0	0
10	CST0200		1	3	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS DRIVER: CREATE_CODELISTFROMCRTDDS	Info	0	0
11	CST0200		1	4	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS DATE: 2014-11-24T11:43:52	Info	0	0
12	CST0200		1	5	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS TYPE: METADATA DERIVATION	Info	0	0
13	CST0200		1	6	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS SASREFERENCES: work._cstsasrefs	Info	0	0
14	CST0200		1	7	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS STUDYROOTPATH: C:\cstSampleLibrary/cdisc-crtdds-1.0-1.7	Info	0	0
15	CST0200		1	8	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS GLOBALLIBRARY: C:\cstGlobalLibrary	Info	0	0
16	CST0200		1	9	SDTMUTIL_CREATEFORMATSFROMCRTDDS	PROCESS CSTVERSION: 1.7	Info	0	0
17	CST0200		1	10	SDTMUTIL_CREATEFORMATSFROMCRTDDS	Process completed successfully	Info	0	0
18	CST0102		1	1	CSTUTIL_SAVERESULTS	results.codelist_results was created as requested	Info	0	0

- The data set contains 18 records. One of the last records reports **Process completed successfully**.

  Note: Values that refer to temporary directories, files, or **PROCESS DATE:** vary.

6 Ensure that the ***CST_SDTM*/derived/formats** directory contains a cterms catalog (cterms.sas7bcat).

7 Open the cterms catalog and verify that it has at least 46 formats.

  Note: The data set can show a different number of formats if it previously existed. In this case, the 46 formats are appended to the file.

8 Close the SAS session.

Sign-Off

Test 4: Build SAS Formats
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

21

Test 5: Report Check Metadata

Introduction

This test verifies that all metadata about SDTM 3.1.3 validation checks is properly installed. A sample report itemizes this metadata.

Steps

1 Start a new SAS session.

2 In the SAS Program Editor, select **File** ▶ **Open Program**, and then select ***CST_SDTM*/programs/cst_metadatareport.sas**.

3 Select **Run** ▶ **Submit**.

This program writes to the SAS log file and generates a PDF file named cstcheckmetadatareport.pdf in the ***CST_SDTM*/results** directory.

Note: No result data set is created.

4 Review the log to ensure that there are no errors or warnings.

5 Review the PDF file.

SAS Clinical Standards Toolkit 1.7
CDISC-SDTM 3.1.3 Validation Check Metadata

Check Overview

Validation Check Identifier	Version of Standard	Source of Check	Record Identifier used by Check Source	Rule Description from Checksource	Severity of Check	Domains/Data Sets to which Check Applies	Columns to which Check Applies
SDTM0004	***	SAS	SAS0033	Source metadata includes domain data set not found in reference metadata	Note	_ALL_	
SDTM0005	***	SAS	SAS0034	Custom domain data set does not adhere to specification naming guidelines	Note	_ALL_	
SDTM0006	***	SAS	SAS0035	Source data library contains domain data not found in study metadata	Warning	_ALL_	
SDTM0011	***	WebSDM	IR5250	Identifies a column that was described in the domain description but not included in the SAS dataset for that domain	Note	_ALL_	
SDTM0014	***	SAS	SAS0008	Identifies a column listed in the domain description as Permissible ('Perm') but not included in the SAS dataset for that domain	Note	_ALL_	
SDTM0022	***	SAS	SAS0001	Column length < length defined in standard	Note	_ALL_	
SDTM0023	***	SAS	SAS0002	Column length > length defined in standard	Error	_ALL_	
SDTM0031	***	SAS	SAS0004	Column format found but column not subject to controlled terminology	Error	_ALL_	
SDTM0032	***	SAS	SAS0005	Column format found but format name mismatch with standard controlled terminology name	Note	_ALL_	
SDTM0202	***	SAS	SAS0015	Identifies a null (empty) value found in a column where (Standard) Core attribute is 'Exp'	Note	_ALL_	_ALL_

a Ensure that all four of these report sections were generated:

- The Report Procedure (Check Overview)
- Additional Check Details
- Message Details
- Reference Information

b Ensure that all titles, footnotes, column headings, and cell contents appear correct.

c In the **Reference Information** section, look for at least one value of **WebSDM** in the column named **Source of Information** (for example, the row for validation check SDTM0011).

6 Close the SAS session.

Sign-Off

Test 5: Report Check Metadata
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments