SAS® Clinical Standards Toolkit 1.5

Operational Qualification

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

SAS® Clinical Standards Toolkit 1.5: Operational Qualification

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

Part 1

Before You Begin

2

1

Introduction

Purpose

Starting with the 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-1.5/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 30 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 39 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 7, “Internal Validation,” in the *SAS Clinical Standards Toolkit: User’s Guide*.

This document explains how to verify that the SAS Clinical Standards Toolkit 1.5 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.

Assumptions and Notes

General Assumptions

- The second maintenance release for SAS 9.3 has been installed and is functioning correctly. It is not being tested.
- The SAS Clinical Standards Toolkit 1.5 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.

File Path Separator

This document is used for both the 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.

SASINSTALL and SASROOT within This Document

- SASINSTALL is used to denote the SAS installation directory. This is the directory into which all SAS products are installed.

  The default value in SAS 9.3 on Microsoft Windows is **`C:/Program Files/SASHome`**.

  The default value varies on UNIX computers. Please consult your system administrator.

- SASROOT is used to denote the root directory for the SAS System installation.

  The default value in SAS 9.3 is **`SASINSTALL/SASFoundation/9.3`**.

- *sample study library directory* is used to denote the sample study libraries available with the SAS Clinical Standards Toolkit 1.5.

  The default value for the SAS Clinical Standards Toolkit 1.5 on Microsoft Windows is **C:/cstSampleLibrary**.

Variables Referred to by the Tests

The tests refer to the following 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*/cdisc-sdtm-3.1.1-1.5/sascstdemodata**

- *CST_SDTM*

  ***sample study library directory*/cdisc-sdtm-3.1.2-1.5/sascstdemodata**

- *CST_ODM_130*

  ***sample study library directory*/cdisc-odm-1.3.0-1.5**

- *CST_ODM*

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

- *CST_CRTDDS*

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

Generation of a PDF File

The last manual test (see Chapter 16, “Test 5: Report Check Metadata,” on page 79) 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-1.5/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 30 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, CDSIC 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**.

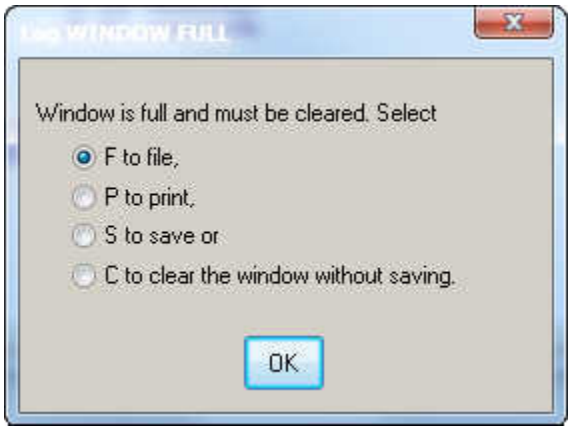
3 Select **Run** ▶ **Submit**.

The program outputs to the SAS log and creates a cstrslt.validation_results data set under ***CST_FRAMEWORK*/results**.

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.

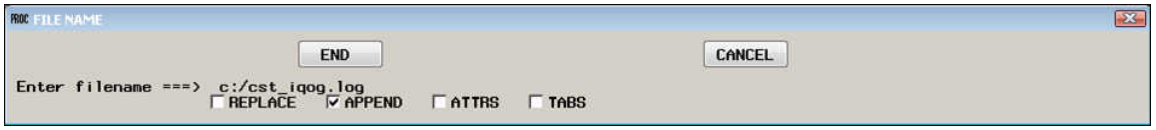
4 If the SAS log reaches it limit, perform these steps:

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 see whether there are any errors or warnings.

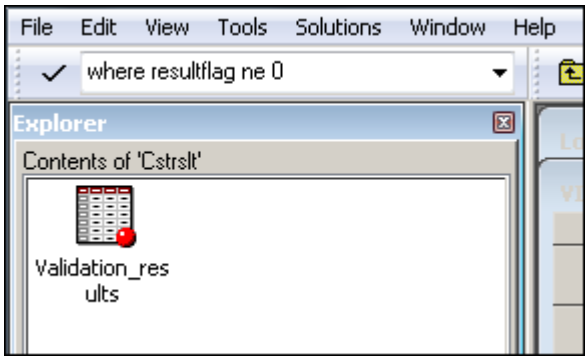
There should be no errors or warnings.

6 The column labeled **Process status** (named _cst_rc) is `0` for all records.

7 Review the cstrslt.validation_results data set using the SAS explorer, especially for the following conditions:

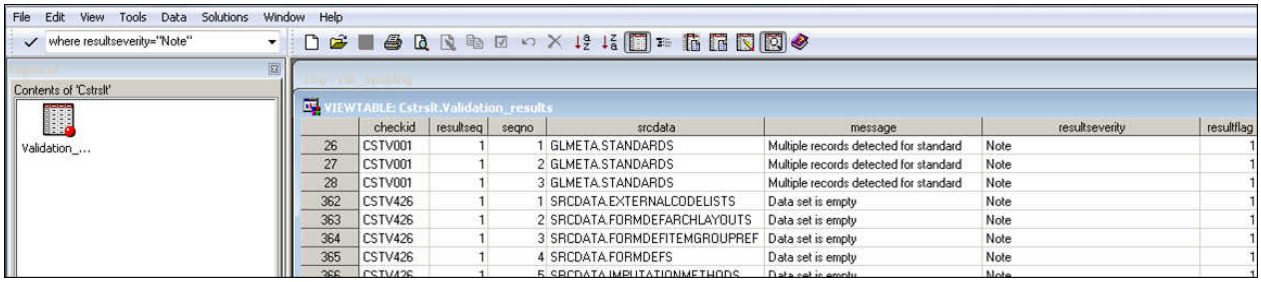
a The field **resultflag** is not 0.

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**.

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



	checkid	resultseq	seqno	srcdata	message	resultseverity	resultflag
26	CSTV001	1	1	GLMETA.STANDARDS	Multiple records detected for standard	Note	1
27	CSTV001	1	2	GLMETA.STANDARDS	Multiple records detected for standard	Note	1
28	CSTV001	1	3	GLMETA.STANDARDS	Multiple records detected for standard	Note	1
362	CSTV426	1	1	SRCDATA.EXTERNALCODELISTS	Data set is empty	Note	1
363	CSTV426	1	2	SRCDATA.FORMDEFARCHLAYOUTS	Data set is empty	Note	1
364	CSTV426	1	3	SRCDATA.FORMDEFITEMGROUPREF	Data set is empty	Note	1
365	CSTV426	1	4	SRCDATA.FORMDEFS	Data set is empty	Note	1

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

- In this example, multiple records are detected because there are multiple standard versions for ODM (1.3.0 and 1.3.1) and SDTM (3.1.1, 3.1.2, and 3.1.3). If multiple records were found for the same standard version, this check would be in error.

	checkid	seqno	srcdata	message	resultseverity	resultflag	actual	keyvalues
26	CSTV001	1	GLMETA.STANDARDS	Multiple records detected for standard	Note	1	keys=standard mnemonic	standard=CDISC-ODM,mnemonic=ODM
27	CSTV001	2	GLMETA.STANDARDS	Multiple records detected for standard	Note	1	keys=standard mnemonic	standard=CDISC-SDTM,mnemonic=SDTM
28	CSTV001	3	GLMETA.STANDARDS	Multiple records detected for standard	Note	1	keys=standard mnemonic	standard=CDISC-SDTM,mnemonic=SDTM

- In this example, 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.

	checkid	seqno	srcdata	message	resultseverity	resultflag	actual	keyvalues
124	CSTV251	1	[CSTMETA.STANDARDS][GLMETA.ST	Check not run, not applicable to this standard	Info	-1		
127	CSTV252	1	[CSTMETA.STANDARDSASREFEREN	Check not run, not applicable to this standard	Info	-1		
128	CSTV252	1	[CSTMETA.STANDARDSASREFEREN	Check not run, not applicable to this standard	Info	-1		

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

	checkid	seqno	srcdata	message	resultseverity	resultflag	actual	keyvalues
136	CSTV262	1	CSTVALIDATE	Check not run, checkstatus < 1	Info	-1	checkstatus=-2 (not implemented in this release)	

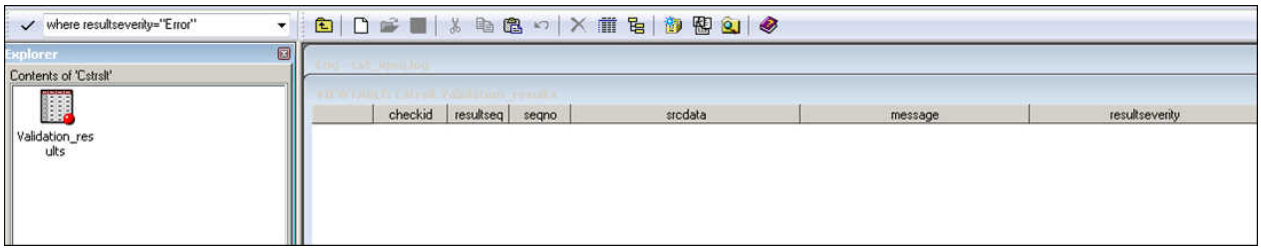
- In this example, these data sets are empty. They are empty because they are templates and do not contain observations.

	checkid	seqno	srcdata	message	resultseverity	resultflag	actual	keyvalues
362	CSTV426	1	SRCDATA.EXTERNALCODELISTS	Data set is empty	Note	1		
363	CSTV426	2	SRCDATA.FORMDEFARCHLAYOUTS	Data set is empty	Note	1		
364	CSTV426	3	SRCDATA.FORMDEFITEMGROUPREF	Data set is empty	Note	1		
365	CSTV426	4	SRCDATA.FORMDEFS	Data set is empty	Note	1		
366	CSTV426	5	SRCDATA.IMPUTATIONMETHODS	Data set is empty	Note	1		
367	CSTV426	6	SRCDATA.ITEMALIASES	Data set is empty	Note	1		

- In this example, **resultseverity** equals **Info** because the controlled terminology does not have an sl_cntl folder associated with it. There are no control type data sets associated with controlled terminology.

	checkid	seqno	srcdata	message	resultseverity	resultflag	actual	keyvalues
1344		3	_CSTREADSTDS	SAS DATASET (sl_cntl.stdvalidation_sasrefs) does not exist	Info	1		

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



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 has ensured 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 1.3.0 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 outputs to the SAS log and creates data sets in the formats, metadata, and data subdirectories in the ***CST_ODM*/derived** directory. It also creates a read_results data set in the ***CST_ODM*/results** directory.

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

There should be no errors or warnings.

5 Review the read_results data set in the ***CST_ODM*/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 message column contains correct paths and process metadata.
- The column labeled **Process status** (named _cst_rc) is `0` for all records.
- The data set contains 54 records (for ODM 1.3.1).
- Row 53 reports that the ODM file was read successfully.

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

   - There are three new SAS data sets and three new SAS format catalogs. (Do not count any data sets that are not SAS, such as .xpt files.)
   - The odmfmtcat_en data set contains 957 records and five columns.

9 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 outputs to the SAS log 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 see whether there are any errors or warnings.

There should be no errors or warnings.

5 Review the validation_results data set in the ***CST_ODM*/results** directory to ensure that the following 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 the following 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 outputs to the SAS log, creates an XML file odm_sample_out.xml in the ***CST_ODM*/sourcexml** directory. It also creates a write_results data set in the ***CST_ODM*/results** directory.

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

There should be no errors or warnings.

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

> **TIP** In the SAS Explorer, you can also 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 that contains **scrdata="ODM_WRITE"** that reports that the ODM file was created.
- The last record that contains **srcdata="ODM_XMLVALIDATE"** reports that no errors were found in the ODM file.

6 Ensure that the **CST_ODM/sourcexml** directory contains a new XML file odm_sample_out.xml that has the same size (321 KB) as the XML file odm_sample.xml in the same directory.

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 outputs to the SAS log and creates a readxmltags_results data set in the ***CST_ODM*/results** directory.

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

There should be no errors or warnings.

5 Review the readxmltags _results data set in the ***CST_ODM*/results** directory to ensure that the following 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 outputs to the SAS log and creates a validation_results data set and a validation_metrics data set in the ***CST_CRTDDS*/results** directory.

Note: This code 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 see whether there are any errors or warnings.

There should be no errors or warnings.

5 Review the validation_results data set in the ***CST_CRTDDS*/results** directory to ensure that the following 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"`**.

6 Review the validation_metrics data set in the ***CST_CRTDDS*/results** directory to ensure that the following 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

Introduction

If this program runs successfully and produces the expected results, the SAS Clinical Standards Toolkit has 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, and submit the sample program ***CST_CRTDDS/programs/create_crtdds_from_sdtm.sas***.

   The sample program outputs to the SAS Log and creates 39 data sets in the ***CST_CRTDDS/data*** directory and a Results data set in the ***CST_CRTDDS/results*** directory.

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

There might be one warning when using the SAS Clinical Standards Toolkit sample data:

```
WARNING: Multiple lengths were specified for the variable Role
by input data set(s). This may cause truncation of data.
```

3 Review the ***CST_CRTDDS*/data** directory to ensure that the following 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.

4 Review the sdtmtodefine_results data set in the ***CST_CRTDDS*/results** directory to ensure that the following 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.

5 Close the SAS session.

Sign-Off

Test 2: Create SAS CRT-DDS from SDTM
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 the 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_sascrtdds_fromxml.sas**.

3 Select **Run** ▶ **Submit**.

This program outputs to the SAS log and creates the SAS representation of the CRT-DDS data sets in the ***CST_CRTDDS*/`derivedata`** directory.

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

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

There should be no errors or warnings.

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

> **TIP** In the SAS Explorer, you can also 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 50 records.
- There is a record that contains `scrdata="CRTDDS_READ"` that reports that the define.xml file was read successfully.
- There is a record that contains `srcdata="JAVA CHECK"` that reports `"No java issues"`.

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

7 Open the clitemdecodetranslatedtext SAS data set.

It must contain 2909 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	N80052
2	DOSE NOT CHANGED	en	N80063
3	DOSE REDUCED	en	N80074
4	DRUG INTERRUPTED	en	N80085
5	DRUG WITHDRAWN	en	N80096
6	NOT APPLICABLE	en	N80107
7	UNKNOWN	en	N80118
8	MILD	en	N80136
9	MODERATE	en	N80147
10	SEVERE	en	N80158
11	DAYS	en	N80176
12	HOURS	en	N80187
13	MONTHS	en	N80198
14	WEEKS	en	N80209
15	YEARS	en	N80220
16	ABW	en	N80238
17	AFG	en	N80249

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: Import from XML, Export to XML (Round Trip)

Introduction

This test reads a CRT-DDS define.xml file and creates the 39 SAS data sets in the Work library representing the CRT-DDS 1.0 model. It then exports these generated data sets and creates a CRT-DDS define.xml file. This verifies the round-tripping from XML to data to XML.

Steps

1 Start a new SAS session.

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

3 Select **Run** ▶ **Submit**.

This program outputs to the SAS log and creates a define1-0-0.xsl file and a define_export.xml file in the ***CST_CRTDDS*/sourcexml** directory. It also creates a import_results data set in the ***CST_CRTDDS*/results** directory.

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

There should be no errors or warnings.

5 In the SAS Explorer, review the data in the **Work** directory to ensure that the following conditions are met:

- There are 39 data sets representing SAS interpretation of the CRT-DDS model.
- These data sets do not contain any underscores in their names.

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

> **TIP** In the SAS Explorer, you can view it as **import_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 47 records.
- There is a record that contains `scrdata="CRTDDS_READ"` that reports that the define_import.xml file was read successfully.
- There is a record that contains `srcdata="JAVA CHECK"` that reports `"No java issues"`.

7 Review the export_results data set in the ***CST_CRTDDS*/results** directory to ensure that the following conditions are met:

> **TIP** In the SAS Explorer, you can view it as **export_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 105 records.
- There is a record that contains `scrdata="CRTDDS_WRITE"` that reports that the define_export.xml file was created (row 82).
- There is a record that contains `srcdata="JAVA CHECK"` that reports `"No java issues"` (row 85).

8 In the *`CST_CRTDDS`*`/sourcexml` directory, ensure that the files define_import.xml and define_export.xml are each 201 KB.

9 Double-click the define_export.xml file to open it, and click **Vital Signs** next to the VS table.

The VS table appears.

10 Click the **VSTESTCD** variable.

11 Click the **SIZE** variable in the **VSTESTCD–FRMSIZE** row, and ensure that the table looks like this:

SIZE, Reference Name (SIZE)	
SMALL	SMALL
MEDIUM	MEDIUM
LARGE	LARGE

12 Close the SAS session.

Sign-Off

Test 4: Import from XML, Export to XML (Round Trip)
Signature
Date test was executed
Did the test pass? (Yes or No)
Comments

11

Test 5: Create Define.xml

Introduction

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

Steps

1 Start a new SAS session, and submit the sample program ***CST_CRTDDS/programs/create_crtdds_define.sas***.

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

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

There should be no errors or warnings.

3 Ensure that two files were generated in the ***CST_CRTDDS*/sourcexml** directory: define.xml and define1-0-0.xsl.

4 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 define1-0-0.xsl to an environment where you can display the define.xml file in a web browser.

Note: The style sheet information in define1-0-0.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.

5 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.

6 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

7 In the last table in define.xml (which contains controlled terminology), ensure that the last few items in the file are values for VSTESTCD, including the values BMI and WEIGHT.

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

Coded Value	Decode
ABSKNF	ABSKNF
BMI	BMI
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
TEMP	TEMP
TRSKNF	TRSKNF
WEIGHT	WEIGHT

8 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

SDTM

12

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 outputs to the SAS log and generates a validation_results data set and a validation_metrics data set.

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

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

There should be no errors or warnings.

5 Review the validation_results data set to ensure that the following 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. There are not any checks with the result `severity = 'Warning: Check not run'`.
- The data set contains 312 records.

6 Review the validation_metrics data set and ensure that it contains these last few rows:

Metric Parameter	Count of Records
# of distinct check invocations	15
# check invocations not run	1
Errors (severity=High) reported	1
Warnings (severity=Medium) reported	91
Notes (severity=Low) reported	118
Structural errors, warnings and notes	0
Content errors, warnings and notes	210

7 Close the SAS session.

Sign-Off

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

13

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 code outputs to the SAS log 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 the following 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'`**.

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)
CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibrary/standards/cst-frame	Info	0	0
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
CST0102		1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences was created as requested	Info	0	0
CST0200		1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\geligh\AppData\Local\Temp\Temporary Files_TD5488_L73859_/sasreferences	Info	0	0
CST0200		1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info	0	0
CST0200		1	2	CSTUTIL_ALLOCATESASREFERENCE	SASReferences data set was successfully validated	Info	0	0
CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibrary/standards/cdisc-sdtm	Info	0	0
CST0200		1	1	SDTMUTIL_CREATESASDATAFROMX	PROCESS STANDARD: CDISC-SDTM	Info	0	0
CST0200		1	2	SDTMUTIL_CREATESASDATAFROMX	PROCESS STANDARDVERSION: 3.1.3	Info	0	0
CST0200		1	3	SDTMUTIL_CREATESASDATAFROMX	PROCESS DRIVER: CREATE_SASDATAFROMXPT	Info	0	0
CST0200		1	4	SDTMUTIL_CREATESASDATAFROMX	PROCESS DATE: 2013-04-11T13:39:44	Info	0	0
CST0200		1	5	SDTMUTIL_CREATESASDATAFROMX	PROCESS TYPE: DATA DERIVATION	Info	0	0
CST0200		1	6	SDTMUTIL_CREATESASDATAFROMX	PROCESS SASREFERENCES: work._cstsasrefs	Info	0	0
CST0200		1	7	SDTMUTIL_CREATESASDATAFROMX	PROCESS STUDYROOTPATH: c:/cstSampleLibrary/cdisc-crtdds-1.0-1.5	Info	0	0
CST0200		1	8	SDTMUTIL_CREATESASDATAFROMX	PROCESS GLOBALLIBRARY: c:/cstGlobalLibrary	Info	0	0
CST0200		1	9	SDTMUTIL_CREATESASDATAFROMX	PROCESS CSTVERSION: 1.5	Info	0	0
CST0200		1	10	SDTMUTIL_CREATESASDATAFROMX	Process completed successfully	Info	0	0
CST0102		1	1	CSTUTIL_SAVERESULTS	results.xpt_results was created as requested	Info	0	0

- The data set contains 18 records and that a record near the end reports **`Process completed successfully`**.

  See the sample output above.

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

6 Review the ***`CST_SDTM`*`/derived/data`** directory to ensure that the following 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

14

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 outputs to the SAS log and creates data sets in both the **`results`** and **`derived/metadata`** subdirectories in the ***`CST_SDTM`*** 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.

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

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

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)
CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibrary/standards/cst-framework-1.5/progr	Info	0	0
CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to c:/cstGlobalLibrary/standards/cst-framework-1.5/templ to perform the template lookup	Info	0	0
CST0102		1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences was created as requested	Info	0	0
CST0200		1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\geligh\AppData\Local\Temp\SAS Temporary Files_TD5720_L73859_/sasreferences	Info	0	0
CST0200		1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info	0	0
CST0200		1	2	CSTUTIL_ALLOCATESASREFERENCE	SASReferences data set was successfully validated	Info	0	0
CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibrary/standards/cdisc-sdtm-3.1.3-1.5/pro	Info	0	0
CST0200		1	1	SDTMUTIL_CREATESRCMETAFROMC	PROCESS STANDARD: CDISC-SDTM	Info	0	0
CST0200		1	2	SDTMUTIL_CREATESRCMETAFROMC	PROCESS STANDARDVERSION: 3.1.3	Info	0	0
CST0200		1	3	SDTMUTIL_CREATESRCMETAFROMC	PROCESS DRIVER: CREATE_SOURCEMETADATA	Info	0	0
CST0200		1	4	SDTMUTIL_CREATESRCMETAFROMC	PROCESS DATE: 2013-04-11T13:49:16	Info	0	0
CST0200		1	5	SDTMUTIL_CREATESRCMETAFROMC	PROCESS TYPE: METADATA DERIVATION	Info	0	0
CST0200		1	6	SDTMUTIL_CREATESRCMETAFROMC	PROCESS SASREFERENCES: work._cstsasrefs	Info	0	0
CST0200		1	7	SDTMUTIL_CREATESRCMETAFROMC	PROCESS STUDYROOTPATH: c:/cstSampleLibrary/cdisc-crtdds-1.0-1.5/deriveddata	Info	0	0
CST0200		1	8	SDTMUTIL_CREATESRCMETAFROMC	PROCESS GLOBALLIBRARY: c:/cstGlobalLibrary	Info	0	0
CST0200		1	9	SDTMUTIL_CREATESRCMETAFROMC	PROCESS CSTVERSION: 1.5	Info	0	0
CST0074		1	10	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemod	Info	0	0
CST0074		1	11	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemod	Info	0	0
CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref tmplt was allocated to c:/cstGlobalLibrary/standards/cdisc-sdtm-3.1.3-1.5/tem to perform the template lookup	Info	0	0
CST0102		1	2	CST_CREATEDSFROMTEMPLATE	trgmeta.source_study was created as requested	Info	0	0
CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemod	Info	0	0
CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref tmplt was allocated to c:/cstGlobalLibrary/standards/cdisc-sdtm-3.1.3-1.5/tem to perform the template lookup	Info	0	0
CST0102		1	2	CST_CREATEDSFROMTEMPLATE	trgmeta.source_values was created as requested	Info	0	0
CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemod	Info	0	0
CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref tmplt was allocated to c:/cstGlobalLibrary/standards/cdisc-sdtm-3.1.3-1.5/tem to perform the template lookup	Info	0	0
CST0102		1	2	CST_CREATEDSFROMTEMPLATE	trgmeta.source_documents was created as requested	Info	0	0
CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemod	Info	0	0
CST0102		1	1	CSTUTIL_SAVERESULTS	results.srcmeta_results was created as requested	Info	0	0

- Ensure that the data set contains 28 records.

  See the sample output above.

  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`**.

	Result identifier	Validation check identifier	Unique invocation of resultid	Sequence number within resultseq	Source data	Resolved message text from message file
17	CST0074		1	10	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemodata\derived\metadata
18	CST0074		1	11	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemodata\derived\metadata
21	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemodata\derived\metadata
24	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemodata\derived\metadata
27	CST0074		1	3	SDTMUTIL_CREATESRCMETAFROMC	Study reference data created in c:\cstSampleLibrary\cdisc-sdtm-3.1.3-1.5\sascstdemodata\derived\metadata

6 Review the ***CST_SDTM*****/derived/metadata** directory to ensure that the following 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

15

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 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 code outputs to the SAS log 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**).

At the end of the run, the FMTLIB output appears.

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

There should be no errors or warnings.

5 Review the codelist_results data set in the ***CST_SDTM*/results** directory to ensure that the following 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 the result `severity='Warning: Check not run'`.

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)
CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibrary/standards/cst-framework-1.5/programs/initialize.properties	Info	0	0
CST0200		1	1	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to c:/cstGlobalLibrary/standards/cst-framework-1.5/templates to perform the template lookup	Info	0	0
CST0102		1	2	CST_CREATEDSFROMTEMPLATE	work.sasreferences was created as requested	Info	0	0
CST0200		1	1	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\geligh\AppData\Local\Temp\SAS Temporary Files_TD4068_L73859_/sasreferences	Info	0	0
CST0200		1	1	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info	0	0
CST0200		1	2	CSTUTIL_ALLOCATESASREFERENCE	SASReferences data set was successfully validated	Info	0	0
CST0108		1	1	CST_SETPROPERTIES	The properties were processed from the PATH c:/cstGlobalLibrary/standards/cdisc-sdtm-3.1.3-1.5/programs/initialize.properties	Info	0	0
CST0200		1	1	SDTMUTIL_CREATEFORMATSFROMC	PROCESS STANDARD: CDISC-SDTM	Info	0	0
CST0200		1	2	SDTMUTIL_CREATEFORMATSFROMC	PROCESS STANDARDVERSION: 3.1.3	Info	0	0
CST0200		1	3	SDTMUTIL_CREATEFORMATSFROMC	PROCESS DRIVER: CREATE_CODELISTFROMCRTDDS	Info	0	0
CST0200		1	4	SDTMUTIL_CREATEFORMATSFROMC	PROCESS DATE: 2013-04-11T14:01:09	Info	0	0
CST0200		1	5	SDTMUTIL_CREATEFORMATSFROMC	PROCESS TYPE: METADATA DERIVATION	Info	0	0
CST0200		1	6	SDTMUTIL_CREATEFORMATSFROMC	PROCESS SASREFERENCES: work._cstsasrefs	Info	0	0
CST0200		1	7	SDTMUTIL_CREATEFORMATSFROMC	PROCESS STUDYROOTPATH: c:/cstSampleLibrary/cdisc-crtdds-1.0-1.5	Info	0	0
CST0200		1	8	SDTMUTIL_CREATEFORMATSFROMC	PROCESS GLOBALLIBRARY: c:/cstGlobalLibrary	Info	0	0
CST0200		1	9	SDTMUTIL_CREATEFORMATSFROMC	PROCESS CSTVERSION: 1.5	Info	0	0
CST0200		1	10	SDTMUTIL_CREATEFORMATSFROMC	Process completed successfully	Info	0	0
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 must report `Process completed successfully`.

  See the sample output above.

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 (named cterms.sas7bcat).

7 Open the cterms catalog and verify that it has 53 formats.

Note: The data set can show a different number of formats if it previously existed. In this case, the 53 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

16

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 outputs to the SAS log 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 see whether there are any errors or warnings.

There should be no errors or warnings.

5 Review the PDF file.

SAS Clinical Standards Toolkit 1.5
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
SDTM0001	***	WebSDM	IR5000	Identifies domain table that has zero rows and therefore contains no data	Warning	_ALL_	
SDTM0002	***	SAS	SAS0017	A load of data into JANUS requires that the DM, DS and EX domains be submitted for each study to be loaded.	Error	DM+DS+EX	
SDTM0003	***	SAS	SAS0018	WebSDM and the SDTM model require only the DM domain be present.	Error	DM	
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_	
SDTM0012	***	WebSDM	IR5252	Identifies a column listed in the domain description as Required ('Req') but not included in the SAS dataset for that domain	Error	_ALL_	
SDTM0013	***	WebSDM	IR5253	Identifies a column listed in the domain description as Expected ('Exp') but not included in the SAS dataset for that domain	Warning	_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_	

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 `OpenCDISC` in the column named **Source of Information** (for example, the row for validation check SDTM0231).

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