SAS® Clinical Standards Toolkit 1.6
Operational Qualification
The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2014. SAS® Clinical Standards Toolkit 1.6: Operational Qualification. Cary, NC: SAS Institute Inc.

SAS® Clinical Standards Toolkit 1.6: Operational Qualification
Copyright © 2014, SAS Institute Inc., Cary, NC, USA
All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others’ rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR 227.7202-4 and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government’s rights in Software and documentation shall be only those set forth in this Agreement.


June 2014

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 offerings, visit support.sas.com/bookstore or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.
## PART 5  Define-XML  55

**Chapter 11  /  Test 1: Create Define-XML 2.0 SAS Data Sets from SDTM Source Metadata**  57  
- Introduction  57  
- Steps  57  
- Sign-Off  59

**Chapter 12  /  Test 2: Create Define-XML 2.0 File from SAS Data Sets**  61  
- Introduction  61  
- Steps  61  
- Sign-Off  64

**Chapter 13  /  Test 3: Create SAS Data Sets from Define-XML 2.0 File**  65  
- Introduction  65  
- Steps  65  
- Sign-Off  67

---

## PART 6  SDTM  69

**Chapter 14  /  Test 1: Validate SDTM**  71  
- Introduction  71  
- Steps  71  
- Sign-Off  73

**Chapter 15  /  Test 2: Build Source Data**  75  
- Introduction  75  
- Steps  75  
- Sign-Off  78

**Chapter 16  /  Test 3: Build Source Metadata**  79  
- Introduction  79  
- Steps  79
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Test</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 17</td>
<td>Test 4: Build SAS Formats</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>Steps</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>Sign-Off</td>
<td>87</td>
</tr>
<tr>
<td>Chapter 18</td>
<td>Test 5: Report Check Metadata</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Steps</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Sign-Off</td>
<td>91</td>
</tr>
</tbody>
</table>
Part 1

Before You Begin

Chapter 1

*Introduction*
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-1.6/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 66 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 1.6 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 9.3 or SAS 9.4, has been installed and is functioning correctly.
- The SAS Clinical Standards Toolkit 1.6 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 1.6.
The default value for SAS Clinical Standards Toolkit 1.6 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/cst-framework-1.6

- **CST_SDTM**
  
  sample study library directory/cdisc-sdtm-3.1.3–1.6/
  sascstdemodata

- **CST_ODM**
  
  sample study library directory/cdisc-odm-1.3.1–1.6

- **CST_CRTDDS**
  
  sample study library directory/cdisc-crtdds-1.0–1.6

- **CST_DEFINEXML**
  
  sample study library directory/cdisc-definexml-2.0–1.6

Generation of a PDF File

The last manual test (see Chapter 18, “Test 5: Report Check Metadata,” on page 89) 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.
Chapter 1 / Introduction
Part 2

Internal Validation

Chapter 2

*Installation Qualification and Operational Qualification*
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-1.6/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 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:
A number of observations can have 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 resultflag = "Note" (this is case sensitive) in the control box in the upper left:

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 (3.1.1, 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.

- 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.
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.

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

In this example of where resultseverity="Info", Result severity 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.

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, the following result might be reported:

<table>
<thead>
<tr>
<th>resultid</th>
<th>checkid</th>
<th>resultseq</th>
<th>segno</th>
<th>srcref</th>
<th>message</th>
<th>resultseverity</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>CSTV275</td>
<td>CSTV275</td>
<td>3</td>
<td>14</td>
<td>REFCNTL.VALIDATION_MASTER Data set is empty</td>
<td>Warning</td>
</tr>
</tbody>
</table>

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 SAS Clinical Standards Toolkit 1.6.

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.
<table>
<thead>
<tr>
<th><strong>Sign-Off</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal Validation - Installation Qualification and Operational Qualification</strong></td>
</tr>
<tr>
<td><strong>Signature</strong></td>
</tr>
<tr>
<td><strong>Date test was executed</strong></td>
</tr>
<tr>
<td><strong>Did the test pass? (Yes or No)</strong></td>
</tr>
<tr>
<td><strong>Comments</strong></td>
</tr>
</tbody>
</table>
Part 3

ODM

Chapter 3
Test 1: Create SAS ODM from XML ........................................... 21

Chapter 4
Test 2: Validate SAS ODM ....................................................... 25

Chapter 5
Test 3: Create ODM XML from SAS ODM ................................. 29

Chapter 6
Test 4: Find Unsupported Tags in ODM XML ............................ 33
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 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.
   - A record 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 Close the SAS session.
<table>
<thead>
<tr>
<th><strong>Sign-Off</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test 1: Create SAS ODM from XML</strong></td>
</tr>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
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 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 | SRCDATA.ITEMDEF | The foreign key OID does not have a corresponding value in the target data set SRCDATA.ITEMDEFS | Error | 1 | 0 | CODELISTREF=CodeLists.OID.LETEST |
| 295 | ODM0110 | 110 | SRCDATA.ANNOTATIONFLAG | The foreign key OID does not have a corresponding value in the target data set SRCDATA.ANNOTATIONFLAG | Error | 1 | 0 | FLAGTYPECODELISTOID=CodeLists.OID.dmget.reqLg |

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**

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
<td></td>
</tr>
<tr>
<td>Date test was executed</td>
<td></td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td></td>
</tr>
</tbody>
</table>
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 \texttt{CST\_ODM/source\_xml} directory. It creates a write\_results data set in the \texttt{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 \texttt{CST\_ODM/results} directory to ensure that the following conditions are met:

\textbf{TIP} In the SAS Explorer, you can view it as Write\_results in the Results library.

- The column labeled \textbf{Process status} (named _cst\_rc) is 0 for all records.
- The column named \textbf{resultflag} is 0 for all records.
- The data set contains 70 records.
- There is a record where \textbf{Source data} is \texttt{ODM\_WRITE} that reports that the ODM file was created.
- The last record where \textbf{Source data} contains \texttt{ODM\_XMLVALIDATE} reports that no errors were found in the ODM file.

6 Ensure that the \texttt{CST\_ODM/source\_xml} directory contains a new XML file odm_sample_out.xml that has the same size as the XML file odm_sample.xml in the same directory.

7 Close the SAS session.

\section*{Sign-Off}

\textbf{Test 3: Create ODM XML from SAS ODM}

Signature
Test 3: Create ODM XML from SAS ODM

Date test was executed

Did the test pass? (Yes or No)

Comments
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 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.
<table>
<thead>
<tr>
<th><strong>Sign-Off</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test 4: Find Unsupported Tags in ODM XML</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signature</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Date test was executed</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Did the test pass? (Yes or No)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Comments</th>
</tr>
</thead>
</table>
Part 4

CRT-DDS

Chapter 7
Test 1: Validate CRT-DDS ............................................. 39

Chapter 8
Test 2: Create SAS CRT-DDS from SDTM Metadata ............ 43

Chapter 9
Test 3: Create SAS CRT-DDS from define.xml .................. 47

Chapter 10
Test 4: Create define.xml ............................................. 51
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 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".
     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 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.
<table>
<thead>
<tr>
<th>Sign-Off</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test 1: Validate CRT-DDS</strong></td>
</tr>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
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 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.

6 Review the `sdmtmto define_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 `sdmtmto define_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
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/deriveddata directory.
4. In the SAS Explorer, you can view it in the **srcdata** library.

5. Review the log to see whether there are any errors or warnings. There should be no errors or warnings.

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

   - 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**.

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

8. Open the clitemdecodetranslatedtext SAS data set.

   It contains 4481 observations. The first 17 observations are shown in this figure.
<table>
<thead>
<tr>
<th></th>
<th>Human-readable text appropriate for a particular language</th>
<th>Natural language or country-specific language variant</th>
<th>Foreign key: CodeListItems.DID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DOSE INCREASED</td>
<td>en</td>
<td>N78024</td>
</tr>
<tr>
<td>2</td>
<td>DOSE NOT CHANGED</td>
<td>en</td>
<td>N78035</td>
</tr>
<tr>
<td>3</td>
<td>DOSE REDUCED</td>
<td>en</td>
<td>N78046</td>
</tr>
<tr>
<td>4</td>
<td>DRUG INTERRUPTION</td>
<td>en</td>
<td>N78057</td>
</tr>
<tr>
<td>5</td>
<td>DRUG WITHDRAWN</td>
<td>en</td>
<td>N78068</td>
</tr>
<tr>
<td>6</td>
<td>NOT APPLICABLE</td>
<td>en</td>
<td>N78079</td>
</tr>
<tr>
<td>7</td>
<td>UNKNOWN</td>
<td>en</td>
<td>N78090</td>
</tr>
<tr>
<td>8</td>
<td>MILD</td>
<td>en</td>
<td>N78108</td>
</tr>
<tr>
<td>9</td>
<td>MODERATE</td>
<td>en</td>
<td>N78119</td>
</tr>
<tr>
<td>10</td>
<td>SEVERE</td>
<td>en</td>
<td>N78130</td>
</tr>
<tr>
<td>11</td>
<td>DAYS</td>
<td>en</td>
<td>N78148</td>
</tr>
<tr>
<td>12</td>
<td>HOURS</td>
<td>en</td>
<td>N78159</td>
</tr>
<tr>
<td>13</td>
<td>MONTHS</td>
<td>en</td>
<td>N78170</td>
</tr>
<tr>
<td>14</td>
<td>WEEKS</td>
<td>en</td>
<td>N78181</td>
</tr>
<tr>
<td>15</td>
<td>YEARS</td>
<td>en</td>
<td>N78192</td>
</tr>
<tr>
<td>16</td>
<td>GBR</td>
<td>en</td>
<td>N78210</td>
</tr>
<tr>
<td>17</td>
<td>USA</td>
<td>en</td>
<td>N78221</td>
</tr>
</tbody>
</table>

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
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\source\xml directory. It creates a Results data set in the \CST_CRTDDS\results directory.

3. Select **Run ▶ Submit**.
4 Review the log to see whether there are any errors or warnings. There should be no errors or warnings.

5 Ensure that two files were generated in the \texttt{CST\_CRTDDS/source\_xml} directory: define.xml and define-v1-updated-html.xsl.

6 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.

\textbf{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.

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

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|l|l|l|}
\hline
\textbf{SDTM Datasets for Study study1} & & & & & \\
\hline
\textbf{Dataset} & 	extbf{Description} & \textbf{Class} & \textbf{Structure} & \textbf{Purpose} & \textbf{Keys} & \textbf{Location} \\
\hline
AE & Adverse Events & Events & One record per adverse event per subject & Tabulation & STUDYID, USUBJID, ADEECOD, AECTSTR & Adverse Events SAS transport file \\
CE & Clinical Events & Events & One record per event per subject & Tabulation & STUDYID, USUBJID, CETERM, CESTTDC & 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, CMSTTDC & Concomitant Medications SAS transport file \\
CO & Comments & Special Purpose Domains & One record per comment per subject & Tabulation & STUDYID, USUBJID, COSID & Comments SAS transport file \\
DA & Drug Accountability & Findings & One record per drug accountability finding per subject & Tabulation & STUDYID, USUBJID, DATESTD, DAATTC & 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, DISDECOD, DISSTTDC & Disposition SAS transport file \\
DV & Protocol Deviations & Events & One record per protocol deviation per subject & Tabulation & STUDYID, USUBJID, DVTERM, DVSTTDC & 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, EGTSTTCD, EGPTNUM, EGPTTPRF, EGPTNUM & ECG Test Results SAS transport file \\
EX & Exposure & Interventions & One record per constant dosing interval per subject & Tabulation & STUDYID, USUBJID, EXTRT, EXSTTDC & Exposure SAS transport file \\
\hline
\end{tabular}
\end{table}

\textbf{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.

8 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

In the Controlled Terminology section of the define.xml file, ensure that the last few items in the file are values for VSTESTCD, including the values BMI and WEIGHT.

<table>
<thead>
<tr>
<th>Coded Value</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSKNF</td>
<td>ABSKNF</td>
</tr>
<tr>
<td>BMI</td>
<td>BMI</td>
</tr>
<tr>
<td>BODYFAT</td>
<td>BODYFAT</td>
</tr>
<tr>
<td>BSA</td>
<td>BSA</td>
</tr>
<tr>
<td>DIABP</td>
<td>DIABP</td>
</tr>
<tr>
<td>FARMCIR</td>
<td>FARMCIR</td>
</tr>
<tr>
<td>FRMSIZE</td>
<td>FRMSIZE</td>
</tr>
<tr>
<td>HDCIRC</td>
<td>HDCIRC</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>HEIGHT</td>
</tr>
<tr>
<td>HIPCIR</td>
<td>HIPCIR</td>
</tr>
<tr>
<td>HR</td>
<td>HR</td>
</tr>
<tr>
<td>KNEEHEEL</td>
<td>KNEEHEEL</td>
</tr>
<tr>
<td>LBM</td>
<td>LBM</td>
</tr>
<tr>
<td>MAP</td>
<td>MAP</td>
</tr>
<tr>
<td>OXYSAT</td>
<td>OXYSAT</td>
</tr>
<tr>
<td>PULSE</td>
<td>PULSE</td>
</tr>
<tr>
<td>PULSEPR</td>
<td>PULSEPR</td>
</tr>
<tr>
<td>RESP</td>
<td>RESP</td>
</tr>
<tr>
<td>SAD</td>
<td>SAD</td>
</tr>
<tr>
<td>SSSKNF</td>
<td>SSSKNF</td>
</tr>
<tr>
<td>SYSBP</td>
<td>SYSBP</td>
</tr>
<tr>
<td>TEMP</td>
<td>TEMP</td>
</tr>
<tr>
<td>TRSKNF</td>
<td>TRSKNF</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>WEIGHT</td>
</tr>
</tbody>
</table>
10 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

Chapter 11
Test 1: Create Define-XML 2.0 SAS Data Sets from SDTM Source Metadata ........................................... 57

Chapter 12
Test 2: Create Define-XML 2.0 File from SAS Data Sets .......... 61

Chapter 13
Test 3: Create SAS Data Sets from Define-XML 2.0 File .......... 65
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 see whether there are any errors or warnings.

There should be no errors or warnings.

5 Review the `CST_DEFINEXML/data` directory to ensure that the following conditions are met:

- 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 the following conditions are met:

- The column labeled **Process status** (named `_cst_rc`) is 0 for all records.
- The data set contains 65 records.

7 Close the SAS session.
## Sign-Off

<table>
<thead>
<tr>
<th>Test 1: Create Define-XML 2.0 SAS Data Sets from SDTM Source Metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
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 Java-related and XML-related libraries 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 see whether there are any errors or warnings.
   There should be 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:
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:

<table>
<thead>
<tr>
<th>COM.VS.VSSTRESU</th>
<th>Standard units consistent with CDISC controlled terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM.SUPPQSCG.QVAL.WC.SUPPQSCG.QVAL.00087</td>
<td>QSMM-CRF Page 13; QSCS-CRF Pages 14; QSCG-CRF Page 17</td>
</tr>
<tr>
<td>COM.SUPPQSCG.QVAL.WC.SUPPQSCS.QVAL.00088</td>
<td>QSMM-CRF Page 13; QSCS-CRF Pages 14; QSCG-CRF Page 17</td>
</tr>
<tr>
<td>COM.SUPPQSCS.QVAL.WC.SUPPQSCS.QVAL.00089</td>
<td>QSMM-CRF Page 13; QSCS-CRF Pages 14; QSCG-CRF Page 17</td>
</tr>
</tbody>
</table>

Go to the [top of the define.xml](#)

9 Review the write_results data set in the `CST_DEFINEXML/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 column named **resultflag** is **0** for all records.
- The data set contains **78** records.
- There is a record where **Source data** is **DEFINE_WRITE** that reports that the XML file was created successfully.
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
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 \texttt{CST_DEFINEXML/deriveddata/cdisc-sdtm-3.1.2} directory from the \texttt{CST_DEFINEXML/sourcexml/define2-0-0-example-sdtm.xml} file. It creates a Results data set in the \texttt{CST_DEFINEXML/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 \texttt{CST_DEFINEXML/deriveddata/cdisc-sdtm-3.1.2} directory to ensure that the following conditions are met:

\textbf{TIP} In the SAS Explorer, you can view it in the \texttt{srcdata} library.

- There are 46 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 423 records and 13 columns.

6 Review the \texttt{read_results} data set in the \texttt{CST_DEFINEXML/results} directory to ensure that the following conditions are met:

\textbf{TIP} In the SAS Explorer, you can view it as \texttt{read_results} in the \texttt{Results} library.

- The column labeled \textit{Process status} (named \texttt{_cst_rc}) is 0 for all records.
- The column named \textit{resultflag} is 0 for all records.
- The data set contains 78 records.
- There is a record where \textit{Source data} is \texttt{DEFINE_READ} that reports that the XML file was read successfully.
- There is a record where \textit{Source data} is \texttt{XML TRANSFORMER} that reports \textit{The document validated successfully}.

7 Close the SAS session.
# Sign-Off

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

<table>
<thead>
<tr>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
Part 6

SDTM

Chapter 14
  Test 1: Validate SDTM .................................................. 71

Chapter 15
  Test 2: Build Source Data ............................................. 75

Chapter 16
  Test 3: Build Source Metadata ...................................... 79

Chapter 17
  Test 4: Build SAS Formats ........................................... 85

Chapter 18
  Test 5: Report Check Metadata ..................................... 89
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 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_SDTM/data directory 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 resultseverity='Warning: Check not run'.
   - The data set contains 312 records.

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

<table>
<thead>
<tr>
<th>Metric Parameter</th>
<th>Count of Records</th>
</tr>
</thead>
<tbody>
<tr>
<td># of distinct check invocations</td>
<td>15</td>
</tr>
<tr>
<td># check invocations not run</td>
<td>1</td>
</tr>
<tr>
<td>Errors (severity=High) reported</td>
<td>1</td>
</tr>
<tr>
<td>Warnings (severity=Medium) reported</td>
<td>91</td>
</tr>
<tr>
<td>Notes (severity=Low) reported</td>
<td>118</td>
</tr>
<tr>
<td>Structural errors, warnings and notes</td>
<td>0</td>
</tr>
<tr>
<td>Content errors, warnings and notes</td>
<td>210</td>
</tr>
</tbody>
</table>

7 Close the SAS session.
## Sign-Off

### Test 1: Validate SDTM

<table>
<thead>
<tr>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
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 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'.

76 Chapter 15 / Test 2: Build Source Data
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 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

<table>
<thead>
<tr>
<th>Test 2: Build Source Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
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 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.
   - The column labeled Process status (named _cst_rc) is 0 for all records.
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**.

<table>
<thead>
<tr>
<th>Result identifier</th>
<th>Validation check identifier</th>
<th>Unique invocation of result</th>
<th>Sequence number within result</th>
<th>Source data</th>
<th>Result severity (e.g., warning, error)</th>
<th>Process status</th>
<th>Resolved message text from message file</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CST0108</td>
<td>1</td>
<td>1</td>
<td>CST_SETPROPERTIES</td>
<td>Info</td>
<td>0</td>
<td>The properties were processed from the PATH C:\vsstGlobal\library\dsddm\3.1.3\templates|</td>
</tr>
<tr>
<td>2</td>
<td>CST0200</td>
<td>1</td>
<td>1</td>
<td>CST_CREATEDFROMTEMPLATE</td>
<td>Info</td>
<td>0</td>
<td>The SAS label exists was allocated to C:vsstGlobal\library\dsddm\3.1.3\templates\templates to perform the template lookup</td>
</tr>
<tr>
<td>3</td>
<td>CST0302</td>
<td>1</td>
<td>1</td>
<td>CSTUTIL_PROCESSSETUP</td>
<td>Info</td>
<td>0</td>
<td>Workset references were created as requested</td>
</tr>
</tbody>
</table>
| 4                 | CST0200                     | 1                          | 1                             | CST_INSERTSTANDARDSASRFES | Info                                   | 0             | Process setup is using this SASReferences: C:呕吐\epg\AppData\LocalTemp\SAS Temporary File\_TDSDF\l73999\asrfes
SASReferences data set was successfully validated |
| 5                 | CST0200                     | 1                          | 2                             | CSTUTIL_ALLOCATESASRFES  | Info                                   | 0             | SASReferences data set was successfully validated |
| 6                 | CST0200                     | 1                          | 1                             | CST_SETPROPERTIES       | Info                                   | 0             | The properties were processed from the PATH C:vsstGlobal\library\dsddm\3.1.3\templates\| |
| 7                 | CST0200                     | 1                          | 1                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS STANDARD: CDISC-SDTM |
| 8                 | CST0200                     | 1                          | 2                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS STANDARDVERSION: 3.1.3 |
| 9                 | CST0200                     | 1                          | 3                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS DRIVER: CREATE_SOURCEMETADATA |
| 10                | CST0200                     | 1                          | 4                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS DATE: 2014-01-15 09:54:41 |
| 11                | CST0200                     | 1                          | 5                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS TYPE: METADATA DERIVATION |
| 12                | CST0200                     | 1                          | 6                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS SASREFERENCES: work_cotsatsets
SASReferences data set was successfully validated |
| 13                | CST0200                     | 1                          | 7                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS STUDYROOTPATH |
| 14                | CST0200                     | 1                          | 8                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata |
| 15                | CST0200                     | 1                          | 9                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | PROCESS CVERSION: 1.6 |
| 16                | CST0200                     | 1                          | 1                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | The SAS label was allocated to C:vsstGlobal\library\dsddm\3.1.3.1.6\templates\templates to perform the template lookup |
| 17                | CST0200                     | 1                          | 2                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | TGMETA source: study was created as requested |
| 18                | CST0200                     | 1                          | 3                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 19                | CST0200                     | 1                          | 1                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | The SAS label was allocated to C:vsstGlobal\library\dsddm\3.1.3.1.6\templates\templates to perform the template lookup |
| 20                | CST0200                     | 1                          | 2                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | TGMETA source: values was created as requested |
| 21                | CST0200                     | 1                          | 3                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 22                | CST0200                     | 1                          | 1                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 23                | CST0200                     | 1                          | 2                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 24                | CST0200                     | 1                          | 3                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 25                | CST0200                     | 1                          | 4                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 26                | CST0200                     | 1                          | 5                             | SDTUTIL_CREATEJSONMETAFROMS | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
| 27                | CST0200                     | 1                          | 1                             | SDTUTIL_SAVERESULTS      | Info                                   | 0             | Study reference data created in C:vsstSampleLibrary\dsddm\3.1.3.1.6\rta\tshaderdata\dev |
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

<table>
<thead>
<tr>
<th><strong>Test 3: Build Source Metadata</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>
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 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 resultseverity='Warning: Check not run'.

<table>
<thead>
<tr>
<th>Result identifier</th>
<th>Validation check identifier</th>
<th>Unique invocation of result</th>
<th>Sequence number within results</th>
<th>Source data</th>
<th>Resolved message text from message file</th>
<th>Result severity (e.g., warning, error)</th>
<th>Problem detected (Otherwise, Yes)</th>
<th>Process status (Never, Aborted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CST0100</td>
<td>1</td>
<td>1</td>
<td>CST_SETPROPERTIES</td>
<td>The properties were processed from the PATH C:\c\GLOBAL\Library\standards\cst-framework-1.6\programs\initialize.properties</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>1</td>
<td>CST_CREATEDSFROMTEMPLATE</td>
<td>The SAS Bible context was allocated to C:\c\GLOBAL\Library\standards\cst-framework-1.6\templates to perform the template lookup</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0102</td>
<td>1</td>
<td>2</td>
<td>CST_CREATEDSFROMTEMPLATE</td>
<td>Work solutions was created as requested</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>1</td>
<td>CST_UTIL_PROCEDURE_SETUP</td>
<td>Process setup is using this SAS References: C:\c\GLOBAL\Library\standards\cst-framework-1.6\templates to perform the template lookup</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>1</td>
<td>CST_UTIL_EXTENSIONS_SASREFF</td>
<td>SAS References data set was successfully validated</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>2</td>
<td>CST_UTIL_ALLOCATESASREFERENCE</td>
<td>SAS References data set was successfully validated</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0108</td>
<td>1</td>
<td>1</td>
<td>CST_SETPROPERTIES</td>
<td>The properties were processed from the PATH C:\c\GLOBAL\Library\standards\cst-framework-1.6\programs\initialize.properties</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>1</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS STANDARD_CDICE_SDTM</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>2</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS STANDARDVERSION 3.1.3</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>3</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS DRIVER: CREATE_CODELISTFROMCSTMDD</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>4</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS_DATE: 2014-01-01T00:00:00</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>5</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS TYPE: METADATA DERIVATION</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>6</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS SASREFERENCES: work_extension</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>7</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS STUDY_PRODUCTPATH C:\c\GLOBAL\Library\cdisc-icr10-1.0-1.6</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>8</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS GLOBAL_LIBRARY.C:\c\GLOBAL\Library</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>9</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>PROCESS CVERSION: 1.6</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0200</td>
<td>1</td>
<td>10</td>
<td>SMTUTIL_CREATEFORMATFROMCSTM</td>
<td>Process completed successfully</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CST0102</td>
<td>1</td>
<td>1</td>
<td>CST_UTIL_SAVERESULT</td>
<td>results.codelist_result was created as requested</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- 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 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**
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 `cstcheckmetadatreport.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.

   ![Table of Check Overview](image)

   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

<table>
<thead>
<tr>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date test was executed</td>
</tr>
<tr>
<td>Did the test pass? (Yes or No)</td>
</tr>
<tr>
<td>Comments</td>
</tr>
</tbody>
</table>