



THE
POWER
TO KNOW.

SAS[®] Clinical Standards Toolkit 1.6

Migration Guide

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

SAS® Clinical Standards Toolkit 1.6: Migration Guide

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.

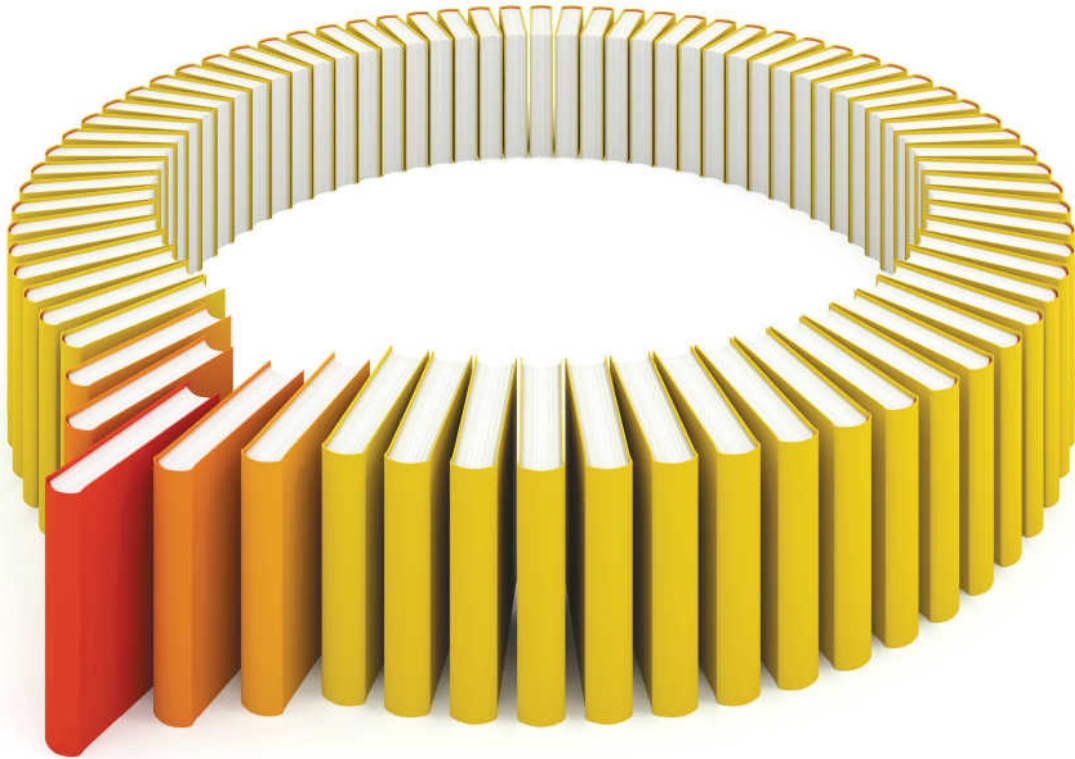
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

May 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.

Contents

<i>Using This Book</i>	<i>vii</i>
Chapter 1 • Installation Locations	1
Introduction	1
!sasroot	1
Global Standards Library	2
Sample Study Library	3
Chapter 2 • SAS Deployment Wizard	5
Introduction to the SAS Deployment Wizard	5
Upgrade to SAS Clinical Standards Toolkit 1.6 in SAS 9.3 (TS1M2) .	6
Upgrade to SAS Clinical Standards Toolkit 1.6 By Installing SAS 9.4 in Conjunction with SAS 9.3 (TS1M2)	9
Chapter 3 • Tools to Identify Differences between Versions	13
Introduction	14
Compare Current User Customizations to the Previous Version of the SAS Clinical Standards Toolkit Using Checksums	16
Compare Two Registered Controlled Terminology Packages	21
Compare Two Codelist Sources	23
Compare Two Macro Autocall Libraries	26
Compare Two Property Files	27
Compare Two Folder Hierarchies	29
Copy a Folder Hierarchy	34
Chapter 4 • Tools to Migrate Source Metadata from CRT-DDS 1.0.0 to Define-XML 2.0.0	37
Introduction	37
Overview	38
Study Source Metadata Migration Driver Programs	39
Chapter 5 • Frequently Asked Questions	45

Installation Issues	46
Migration Process	47
Standards Management	48
Library Issues	48
Other Issues	50
Chapter 6 • Macro API	53
Overview	53

Using This Book

Audience

This document provides guidance on migrating from one version of the SAS Clinical Standards Toolkit to another.

The intended audience is users who have installed and customized SAS Clinical Standards Toolkit 1.5 and now want to install version 1.6.

If you are installing the SAS Clinical Standards Toolkit for the first time and it is version 1.6, or if you have installed previous versions of the SAS Clinical Standards Toolkit but did not customize them, you do not need to review this document or use any of the information when you install the SAS Clinical Standards Toolkit 1.6.

However, if you plan to subsequently customize SAS Clinical Standards Toolkit 1.6, this document offers a preview of the issues that you might want to consider in the future as you migrate to later versions of the SAS Clinical Standards Toolkit.

1

Installation Locations

<i>Introduction</i>	1
<i>!sasroot</i>	1
<i>Global Standards Library</i>	2
<i>Sample Study Library</i>	3

Introduction

The installation process of the SAS Clinical Standards Toolkit puts files in these locations:

- !sasroot
- global standards library
- sample study library

!sasroot

Generic (cross-standard or standard-independent) framework SAS macros are installed in the `!sasroot/cstframework/sasmacro` directory (Microsoft Windows) or in the `!sasroot/sasautos` directory (UNIX).

These primary autocall locations are the same for both SAS 9.3 and SAS 9.4 as well as for both the SAS Clinical Standards Toolkit 1.5 and 1.6. During installation of the SAS Clinical Standards Toolkit 1.6, SAS performs these tasks:

- adds macros that are new to 1.6
- overwrites macros that have been modified in 1.6
- leaves in place macros that have been deprecated with 1.6

CAUTION! If the timestamp of a macro in `!sasroot` is later than the timestamp of the SAS Clinical Standards Toolkit 1.6 macro to be installed, the installation process does not replace the macro in the `!sasroot` location. If you have modified `!sasroot` macros, you assume responsibility for reconciling your changes with newer versions of those macros. Your code management processes must include backing up files so that you do not risk having your changes overwritten when newer macros are installed. There are several tools that help you identify changes in the framework autocall macros.

TIP Best Practice Recommendation: Do not modify the macros in the autocall locations. Make a copy of each macro to modify, modify the copy, and save the copy in another folder. Add the folder to the SAS autocall path.

Global Standards Library

The primary function of the SAS Clinical Standards Toolkit global standards library is to provide the metadata that defines each standard that is supported by the SAS Clinical Standards Toolkit.

The location of the global standards library is specified during installation, but it can be moved to another location. By default, this location is set to the `C:\cstGlobalLibrary` directory (Microsoft Windows) or to the `/usr/local/cstGlobalLibrary` directory (UNIX).

If the global standards library of the SAS Clinical Standards Toolkit 1.5 was installed in the default location, the installation process asks you to specify another location for the

global standards library for 1.6. (For more information, see “[SAS Deployment Wizard](#)” on page 5.) You can rename or move the global standards library for 1.5 before starting the installation of 1.6.

The *global standards library directory/metadata* directory contains the cumulative metadata for every standard that is registered to the SAS Clinical Standards Toolkit. For example, the *metadata/standards.sas7bdat* data set contains 11 records in the SAS Clinical Standards Toolkit 1.5 installed image and 13 records in the 1.6 installed image. The additional records are added support for CDISC-SDTM 3.2 and CDISC-Define-XML 2.0 in 1.6.

If you have customized the SAS Clinical Standards Toolkit 1.5 to add your own standards, you must register those customized standards to the SAS Clinical Standards Toolkit 1.6. This process is described in the “Framework” chapter in the *SAS Clinical Standards Toolkit User’s Guide*.

Sample Study Library

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

The location of the sample study library is specified during installation, but it can be moved to another location. By default, this location is set to the `C:\cstSampleLibrary` directory (Microsoft Windows) or to the `/usr/local/cstSampleLibrary` directory (UNIX).

As a part of each standard definition that is supplied by SAS, the *global standards library directory/standards/standard and version/control/standards.sas7bdat* data set contains a column named `studylibraryrootpath`. By default, this column has been set to the root path of the sample study for that standard. You can use this column to point to the root path of another study hierarchy within your organization. If you have modified this path for any of the standards supplied by SAS, you must ensure that mapping is retained after migrating from the SAS Clinical Standards Toolkit 1.5 to 1.6.

SAS Deployment Wizard

<i>Introduction to the SAS Deployment Wizard</i>	5
<i>Upgrade to SAS Clinical Standards Toolkit 1.6 in SAS 9.3 (TS1M2)</i>	6
Overview	6
Uninstall the SAS Clinical Standards Toolkit 1.5	7
Install the SAS Clinical Standards Toolkit 1.6 for SAS 9.3 (TS1M2)	8
<i>Upgrade to SAS Clinical Standards Toolkit 1.6 By Installing SAS 9.4 in Conjunction with SAS 9.3 (TS1M2)</i>	9
Overview	9
Install the SAS Clinical Standards Toolkit 1.6 for SAS 9.3 (TS1M2)	9

Introduction to the SAS Deployment Wizard

This chapter describes the installation of the SAS Clinical Standards Toolkit 1.6 on Windows 7 Enterprise x64 (Service Pack 1) using the SAS Deployment Wizard in two different scenarios:

- Upgrade to SAS Clinical Standards Toolkit 1.6 in SAS 9.3 (TS1M2)

- Upgrade to SAS Clinical Standards Toolkit 1.6 by installing SAS 9.4 in conjunction with SAS 9.3 (TS1M2) that already has the SAS Clinical Standards Toolkit 1.5 installed

Note: Usually, you use only one of these scenarios, depending on your situation.

Upgrade to SAS Clinical Standards Toolkit 1.6 in SAS 9.3 (TS1M2)

Overview

If you have modified any file that is part of the SAS Clinical Standards Toolkit 1.5, your code management processes should include file backups. File backups minimize these risks:

- A modified file is deleted when you uninstall the SAS Clinical Standards Toolkit 1.5.
- A modified file is overwritten when you install the SAS Clinical Standards Toolkit 1.6.

SAS recommends that you uninstall the SAS Clinical Standards Toolkit 1.5 in SAS 9.3 (TS1M2) before installing 1.6 in SAS 9.3 (TS1M2).

If you do not uninstall the SAS Clinical Standards Toolkit 1.5 and do not upgrade to SAS 9.3 (TS1M2), you risk having the following files that contain deprecated macros (that are not part of the SAS Clinical Standards Toolkit 1.6) left behind in the `!sasroot/cstframework/sasmacro` directory:

- `cst_createds.sas`
- `cst_createemptytables.sas`
- `cst_getstandardmetadatas.sas`
- `cstcheck_java.sas`
- `cstutil_createunixsubdir.sas`
- `csutilwriteresultsintro.sas`

CAUTION! Driver programs in the SAS Clinical Standards Toolkit 1.5 that call these macros and that are migrated to 1.6 might give unexpected results.

Uninstall the SAS Clinical Standards Toolkit 1.5

To uninstall the SAS Clinical Standards Toolkit 1.5, perform the following steps:

- 1 In the Windows Control Panel, uninstall **SAS 9.3**.

The Choose Language dialog box appears.

- 2 Select a language, and then click **OK**.

The Select SAS Products to Uninstall page of the SAS Deployment Wizard appears.

- 3 Select these products:

- SAS Clinical Standards Toolkit Global Standards Library 1.5
- SAS Clinical Standards Toolkit Sample Library 1.5
- SAS Clinical Standards Toolkit Framework JAR File 1.5
- SAS Foundation 9.3

- 4 Click **Next**.

SAS prepares to uninstall the products. The Checking System page appears.

- 5 Click **Next**, and then click **Start**.

Once the uninstall process is complete, the `!sasroot/cstframework/sasmacro` folder is deleted.

- 6 If the `!sasroot/cstframework/sasmacro` folder has not been deleted automatically, delete it manually.

If it has not been deleted, old macros remain in the folder. If you previously updated any of the macros in this folder, the old macros were not deleted.

CAUTION! Ensure that you have backed up the SAS Clinical Standards Toolkit macros that you want to preserve.

- 7 If the global standards library folder and the sample study library folder exist, manually delete them.

These folders might contain previous installation information that is not needed.

The default locations for these folders are `C:\cstGlobalLibrary` and `C:\cstSampleLibrary`.

Install the SAS Clinical Standards Toolkit 1.6 for SAS 9.3 (TS1M2)

To install the SAS Clinical Standards Toolkit 1.6 for SAS 9.3 (TS1M2), perform the following steps:

- 1 Run `setup.exe`, which is located in the new SAS Software Install Depot.

The Choose Language dialog box appears.

- 2 Select a language, and then click **OK**.

The Select Deployment Task page of the SAS Deployment Wizard appears.

- 3 Select **Install SAS software**, and then click **Next**.

The Select Products to Install page appears.

- 4 Select **SAS Foundation**, and then click **Next**.

The Select SAS Foundation Products page appears.

- 5 Select **SAS Clinical Standards Toolkit Framework**, and then click **Next**.

The Specify SAS Installation Data File page appears.

- 6 Enter the path to the SAS installation data file, and then click **Next**.

The Select Language Support page appears.

- 7 Select one or more languages, and then click **Next**.

The Specify SAS Clinical Standards Toolkit Libraries page appears.

- 8 Enter the locations of the global standards library and the sample study library.

Note: The locations cannot be the same. And, the locations cannot already exist; they must be created by the SAS Deployment Wizard.

- 9 Click **Next** twice.

The installation begins.

Upgrade to SAS Clinical Standards Toolkit 1.6 By Installing SAS 9.4 in Conjunction with SAS 9.3 (TS1M2)

Overview

This scenario assumes that you have installed SAS 9.3 (TS1M2) with the SAS Clinical Standards Toolkit 1.5. You can keep this installation and install SAS 9.4 with the SAS Clinical Standards Toolkit 1.6 in a separate environment.

Install the SAS Clinical Standards Toolkit 1.6 for SAS 9.3 (TS1M2)

To install the SAS Clinical Standards Toolkit 1.6 for SAS 9.3 (TS1M2), perform the following steps:

- 1 Run setup.exe, which is located in the new SAS Software Install Depot.

The Choose Language dialog box appears.

- 2 Select a language, and then click **OK**.

The Select Deployment Task page of the SAS Deployment Wizard appears.

- 3 Select **Install SAS software**, and then click **Next**.

The Specify SAS Home page appears because SAS 9.3 (TS1M2) is installed. The SAS Deployment Wizard requires you to specify a different location for SASHome for SAS 9.3 (TS1M2).

4 Enter a location for SASHome.

On Windows 7 Enterprise x64 (Service Pack 1), the default location is `C:\Program Files\SASHome2`.

5 Click **Next**.

The Select Deployment Type page appears.

6 Select **Install SAS Foundation and Related Software**, and then click **Next**.

The Select Products to Install page appears.

7 Select **SAS Foundation**, and then click **Next**.

The Select SAS Foundation Products page appears.

8 Select **SAS Clinical Standards Toolkit Framework**, and then click **Next**.

The Specify SAS Installation Data File page appears.

9 Enter the path to the SAS installation data file, and click **Next**.

The Select Language Support page appears.

10 Select one or more languages, and then click **Next**.

The Select Regional Settings page appears.

11 Specify the regional settings, and then click **Next**.

The Default Product for SAS File Types page appears.

12 Specify the default product to use to open SAS file types, and then click **Next**.

The Specify SAS Clinical Standards Toolkit Libraries page appears.

13 Enter the locations of the global standards library and the sample study library, and then click **Next**.

Note: The locations cannot be the same. And, the locations cannot already exist; they must be created by the SAS Deployment Wizard.

14 Click Next.

The Checking System page appears.

15 Click Next.

The Deployment Summary page appears.

16 Click Start.

The installation begins.

After the installation is complete, the Deployment Complete page appears.

17 Click Next.

The Additional Resources page appears.

18 Click Finish.

3

Tools to Identify Differences between Versions

<i>Introduction</i>	14
<i>Compare Current User Customizations to the Previous Version of the SAS Clinical Standards Toolkit Using Checksums</i>	16
Overview	16
cstutilgeneratechecksums Macro	17
cstutilcomparechecksums Macro	17
compare_checksums.sas Driver Program	18
Examples: Compare Checksum Files for !sasroot	18
Example: Compare Checksum Files for the Global Standards Library	21
Example: Compare Checksum Files for the Sample Study Library	21
<i>Compare Two Registered Controlled Terminology Packages</i>	21
Overview	21
cstutilcompareregisteredct Macro	22
<i>Compare Two Codelist Sources</i>	23
Overview	23
cstutilcomparecodelists Macro	23
<i>Compare Two Macro Autocall Libraries</i>	26
Overview	26
cstutilcompareautocallmacros Macro	26

Compare Two Property Files	27
Overview	27
cstutilcompareproperties Macro	28
Compare Two Folder Hierarchies	29
Overview	29
cstutilcomparefolderhierarchy Macro	29
Example 1: Compare Global Standards Library Folder Hierarchies	30
Example 2: Compare Global Standards Library Files	32
Copy a Folder Hierarchy	34
Overview	34
cstutilcopyfolderhierarchy Macro	34

Introduction

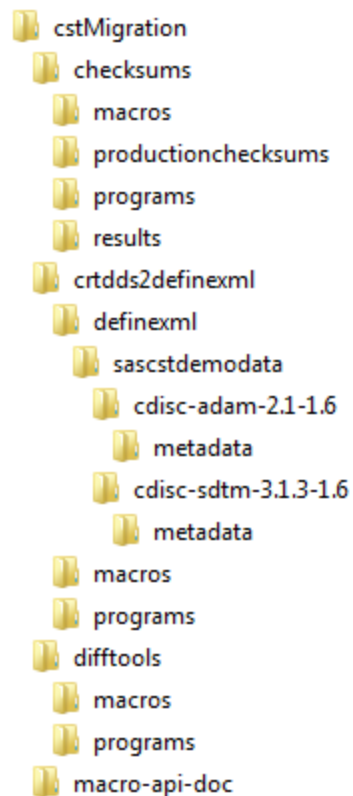
SAS supplies tools to identify and report differences between two entities, such as folders or controlled terminology packages.

Note: Reconciliation of differences is outside the scope of this document.

The tools are contained in the `cstmigration.zip` file, which is located in the “Papers” section at <http://support.sas.com/rnd/base/cdisc/cst/index.html>.

This display shows the folder hierarchy for the migration tools:

Display 3.1 Folder Hierarchy for Migration Tools



Based on additional testing and user feedback, some of these tools might be incorporated into future versions of the SAS Clinical Standards Toolkit.

Compare Current User Customizations to the Previous Version of the SAS Clinical Standards Toolkit Using Checksums

Overview

As described in “[Installation Locations](#)” on page 1, the SAS Clinical Standards Toolkit places files in three locations: !sasroot, the global standards library, and the sample study library. To help you determine whether the files in the installation locations for a previous version of the SAS Clinical Standards Toolkit have been modified from the files that were originally supplied by SAS, SAS supplies checksum files for previous versions.

The cstmigration.zip file contains checksum files for these versions of the SAS Clinical Standards Toolkit:

Table 3.1 Checksum Files for Versions of the SAS Clinical Standards Toolkit

Version	Checksum File From the Root Path \cstMigration\checksums\productionchecksums\
1.5	checksums_cst15_cstframework.xml checksums_cst15_cstgblstdlib.xml checksums_cst15_cstsamplelib.xml
1.5.1	checksums_cst151_cstframework.xml checksums_cst151_cstgblstdlib.xml checksums_cst151_cstsamplelib.xml
1.6	checksums_cst16_cstframework.xml checksums_cst16_cstgblstdlib.xml checksums_cst16_cstsamplelib.xml

cstutilgeneratechecksums Macro

The `cstutilgeneratechecksums` macro creates a checksum file for the specified folder. You usually specify the folder that contains customized files in your version of the SAS Clinical Standards Toolkit. This checksum file is used by the `cstutilcomparechecksums` macro to compare against another checksum file.

This example illustrates the use of the `cstutilgeneratechecksums` macro to create a checksum file for the folder `C:/myCustomGlobalLibrary`:

```
libname custom "<user location for generated checksum XML file>";
%cstutilgeneratechecksums (
  _cstFolder=C:/myCustomGlobalLibrary,
  _cstXMLFile=%sysfunc(pathname(custom))/checksums_cst15_myCustomGlobalLibrary.xml,
  _cstProdCode=cstgblstdlib,
  _cstLabel=CST 1.5 Custom Global Library
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

cstutilcomparechecksums Macro

The `cstutilcomparechecksums` macro compares two checksum files that were created by the `cstutilgeneratechecksums` macro.

This example illustrates the use of the `cstutilcomparechecksums` macro to compare the checksum file created in the previous example to the checksum file for the SAS Clinical Standards Toolkit 1.6 as supplied by SAS:

```
%cstutilcomparechecksums (
  _cstBaseXMLFile=%sysfunc(pathname(chksums))/checksums_cst16_cstlib.xml,
  _cstCompXMLFile=%sysfunc(pathname(chksums))/checksums_cst15_myCustomGlobalLibrary.xml,
  _cstCompResults=work.cstlib,
  _cstOutReportPath=%sysfunc(pathname(chksums)),
  _cstOutReportFile=cstlib_myCustomGlobalLibrary_16.html
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

compare_checksums.sas Driver Program

The `compare_checksums.sas` driver program illustrates using the `cstutilgeneratechecksums` macro and the `cstutilcomparechecksums` macro to compare the checksum files for `!sasroot`, the global standards library, and the sample study library. The checksum files for these versions of the SAS Clinical Standards Toolkit are compared:

- 1.5 to 1.5.1
- 1.5.1 to 1.6
- 1.5 to 1.6

Note: The `cstmigration.zip` file contains the results of the comparisons in the `C:\cstMigration\checksums\results` folder. The results shown in the following examples assume that the installed files have not been customized.

Examples: Compare Checksum Files for !sasroot

This example in the `compare_checksums.sas` driver program compares the checksum files for the `cstframework` autocall library in `!sasroot` of the SAS Clinical Standards Toolkit 1.5 to 1.5.1:

```
%cstutilcomparechecksums (  
  _cstBaseXMLFile=%sysfunc (pathname (chksums) ) /checksums_cst15_cstframework.xml,  
  _cstCompXMLFile=%sysfunc (pathname (chksums) ) /checksums_cst151_cstframework.xml,  
  _cstCompResults=work.cstframework,  
  _cstOutReportPath=&_cstMigrationPath/results,  
  _cstOutReportFile=cstframework_15_151.html  
);
```

The results are written to the location that is specified by the `_cstOutReportPath` parameter.

Note: This location is used by all of the `cstutilcomparechecksums` macro examples.

The `cstutilcomparechecksums` macro creates two lists: one lists the files that are different and one lists the files that are identical.

This is the list of files that are different:

Display 3.2 Different Files between the SAS Clinical Standards Toolkit 1.5 and 1.5.1

Base Directory: C:\Program Files\SASHome\SASFoundation\9.3\cstframework\sasmacro (2014-02-18T16:52:18)
 Compare Directory: C:\Program Files\SASHome\SASFoundation\9.3\cstframework\sasmacro (2014-02-18T17:50:26)
 SAS Production Code: cstframework

Differences in Files

File	Checksum (CST 1.5)	Checksum (CST 1.5.1)	Result
cst_createds.sas	2ce855f0bd8fe91d3cf6482b2ea5f32	79f775bfe8337f731a2b9a1026f82506	Files are different
cstcheckutilfindsasrefsfile.sas	3b0c0f52bc70bd7abd7a4c81b307f0c4	f05a2c39e193ee74db389ef2ebd55a09	Files are different
cstutil_allocatesasreferences.sas	77e1e7b81fb7cf14418c7d8f6f995b31e	b2cbe3fed26a9c0b36b10464a47879f5	Files are different
cstutil_readxmltags.sas	2e54b298472cd83e582a12c6a91e0597	0b637aba477ed512af1cac9322838306	Files are different
cstutil_writecoubexml.sas	4bab05fe582963d73171e51384a767db	c4f6047b28ef5e9a6bf0f0234658100	Files are different
cstutil_writedombexml.sas	fa6944f8d44dda11958bc32300ab1a47	0ec77825894f15ae42d746afd5470a28	Files are different
cstutilbuildmetadatafromsasrefs.sas	86a72549926c748cfa2aaef3b7ae6c9	5cab7f9c43341755e2410d2774ac155d	Files are different
cstutilbuildstdvalidationcode.sas	cb1ed4601454b7b301be043d2501ef06	5b1e6ad1ae55084f7b290f1deaed6c09	Files are different
cstutilcreateattribfromds.sas	a15e1b10c3c9c2880ae131cb7cb1c1f2	e7c7e891134c6b50f0e15d8fc3bb0c28	Files are different
cstutilvalidatesasreferences.sas	e5b35d3d6ef1523f0c1e92fbae450c7	3334fec57c26853c8c601339308778c7	Files are different

Note: The list of files that are identical is not shown.

The compare_checksums.sas driver program also compares the checksum files between major versions. Such a comparison identifies more differences. For example, comparing the cstframework autocall library for SAS Clinical Standards Toolkit 1.5.1 to 1.6 creates three lists: one lists the files that are different, one lists the files that are missing, and one lists the files that are identical.

The list of the files that are different between major versions is longer than the list of the files that are different between minor versions.

Note: Most of the changes to the files that are different between major versions generally involve only the macro headers, not the functionality of the macros. So, the list is not shown.

This is the list of files that are missing between the SAS Clinical Standards Toolkit 1.5.1 and 1.6:

Display 3.3 Files Missing between the SAS Clinical Standards Toolkit 1.5.1 and 1.6

Base Directory: C:\Program Files\SASHome\SASFoundation\9.3\cstframework\sasmacro (2014-02-18T17:50:26)
 Compare Directory: C:\Program Files\SASHome2\SASFoundation\9.4\cstframework\sasmacro (2014-02-18T17:50:30)
 SAS Production Code: cstframework

Missing Files

File	Checksum (CST 1.5.1)	Checksum (CST 1.6)	Result
cst_createdis.sas	79f775bfe8337f731a2b9a1026f62506		File not found in COMP
cst_createemptytables.sas	d55b0b3e516d0d3eb8a9a1355f053402		File not found in COMP
cst_getstandardmetadata.sas	7b33c6281c000d029b3ea36172fb8fb		File not found in COMP
cstcheck_java.sas	32b2cc0a00283e338dfeb5c734d18215		File not found in COMP
cstgetmetadataforstandard.sas		def0799aef0e4c56c6297126e416822	File not found in BASE
cstutil_createunixsubdir.sas	c217b4431c3fb25a5384f43956805f3c9		File not found in COMP
cstutiladddataset.sas		75c002d533ec9b7b0c06cb74077ebf41	File not found in BASE
cstutiladdcolumn.sas		374eb9ecec9641e2b82f27502e7f97	File not found in BASE
cstutilappendmetadatarecords.sas		beedf0417a6351ccc3b77ae411e225dd	File not found in BASE
cstutilbuildattrfromdis.sas		ad93b57d319860f0e2da570100f23f5e	File not found in BASE
cstutilcreatetablesfrommetadata.sas		4de975ca751d7a26faa3c10aa504f172	File not found in BASE
cstutildeletedcolumn.sas		18567dad5924159cb0d177d5a9d8f79d	File not found in BASE
cstutildeletemetatadarecords.sas		94cf12fa6016d4bbe9c2172b708c5731	File not found in BASE
cstutilgetattribute.sas		e5c0bd9e8597b2c5bfa96b41c3d9de662	File not found in BASE
cstutilgetdislook.sas		a852bd50848eb9e9ca0815858509e268	File not found in BASE
cstutillogevent.sas		d31257c79921548692d6cf87d3c5bbee	File not found in BASE
cstutilmodifycolumnattribute.sas		a291ff14ec5734bae4b102d121a7c9f6	File not found in BASE
cstutilprocessxmllog.sas		4b63f333820bd68d3668aa2a3a11f243	File not found in BASE
cstutilsetcstgroot93.sas		ce1d6b997305192cc9c0b39a0bb5644d	File not found in BASE
cstutilsetcstroot93.sas		15d1dde07a12f55a3276645b49d4ffa	File not found in BASE
cstutilupdatemetadatarecords.sas		666188906dc7d9a22e96172056705c8e	File not found in BASE
cstutilwriteresultsintro.sas		1ad33c37b996a99171b396bf1bd0bdae	File not found in BASE
cstutilxmlvalidate.sas		b372f5bed4cdc0471a48f2b9f73b7ffa	File not found in BASE
csutilwriteresultsintro.sas	05dc53a02929485b0b0d7725019a1400		File not found in COMP

This list identifies the files that are no longer in the SAS Clinical Standards Toolkit 1.6 (File not found in COMP) and the files that are new to 1.6 (File not found in BASE).

Note: The list of files that are identical between the two versions is not shown.

Example: Compare Checksum Files for the Global Standards Library

This example in the `compare_checksums.sas` driver program compares checksum files for the global standards library of the SAS Clinical Standards Toolkit 1.51 to 1.6:

```
%cstutilcomparechecksums (
  _cstBaseXMLFile=%sysfunc(pathname(chksums))/checksums_cst151_cstgblstdlib.xml,
  _cstCompXMLFile=%sysfunc(pathname(chksums))/checksums_cst16_cstgblstdlib.xml,
  _cstCompResults=work.cstgblstdlib,
  _cstOutReportPath=&_cstMigrationPath/results,
  _cstOutReportFile=cstgblstdlib_151_16.html
);
```

Example: Compare Checksum Files for the Sample Study Library

This example in the `compare_checksums.sas` driver program compares checksum files for the sample study library of the SAS Clinical Standards Toolkit 1.51 to 1.6:

```
%cstutilcomparechecksums (
  _cstBaseXMLFile=%sysfunc(pathname(chksums))/checksums_cst151_cstlib.xml,
  _cstCompXMLFile=%sysfunc(pathname(chksums))/checksums_cst16_cstlib.xml,
  _cstCompResults=work.cstlib,
  _cstOutReportPath=&_cstMigrationPath/results,
  _cstOutReportFile=cstlib_151_16.html
);
```

Compare Two Registered Controlled Terminology Packages

Overview

To help you determine whether new controlled terminology packages are available with the latest version of SAS Clinical Standards Toolkit, SAS supplies the `cstutilcompareregisteredct` macro.

cstutilcompareregisteredct Macro

The `cstutilcompareregisteredct` macro compares records in one SAS Clinical Standards Toolkit `standardsubtypes` data set with records in another `standardsubtypes` data set (such as in two different versions of the SAS Clinical Standards Toolkit). The default folder for the `standardsubtypes` data set is *global standards library directory/standards/cdisc-terminology-SAS Clinical Standards Toolkit version/control*.

This example compares the SAS Clinical Standards Toolkit 1.6 `standardsubtypes` data set to the 1.5 `standardsubtypes` data set:

```
libname newct 'C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\control';
libname oldct 'C:\cstGlobalLibrary15\standards\cdisc-terminology-1.5\control';

%cstutilcompareregisteredct (
    _cstBaseCT=oldct.standardsubtypes,
    _cstNewCT=newct.standardsubtypes,
    _cstRptType=DATASET,
    _cstRptDS=work.CTchanges,
    _cstOverwrite=Y
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

This display illustrates the data set created by the `cstutilcompareregisteredct` macro (not all columns are shown):

Display 3.4 *Controlled Terminology Package Differences Reported by the `cstutilcompareregisteredct` Macro*

	standardversion	description	message
1	CDISC-CDASH	CDISC CDASH Controlled Terminology, released by NCI on 2011-04-08	Previous CT package found in <code>oldct.standardsubtypes</code> but not found in <code>newct.standardsubtypes</code>
2	CDISC-CDASH	CDISC CDASH Controlled Terminology, released by NCI on 2013-12-20	New CT package found in <code>newct.standardsubtypes</code> but not found in <code>oldct.standardsubtypes</code>
3	CDISC-QS	CDISC QS Controlled Terminology, released by NCI on 2013-12-20	New CT package found in <code>newct.standardsubtypes</code> but not found in <code>oldct.standardsubtypes</code>
4	CDISC-QS	Current QS Controlled Terminology, Copy of 2013-12-20	New CT package found in <code>newct.standardsubtypes</code> but not found in <code>oldct.standardsubtypes</code>
5	CDISC-SDTM	CDISC SDTM Controlled Terminology, released by NCI on 2011-04-08	Previous CT package found in <code>oldct.standardsubtypes</code> but not found in <code>newct.standardsubtypes</code>
6	CDISC-SDTM	CDISC SDTM Controlled Terminology, released by NCI on 2013-12-20	New CT package found in <code>newct.standardsubtypes</code> but not found in <code>oldct.standardsubtypes</code>
7	CDISC-SEND	CDISC SEND Controlled Terminology, released by NCI on 2012-01-03	Previous CT package found in <code>oldct.standardsubtypes</code> but not found in <code>newct.standardsubtypes</code>
8	CDISC-SEND	CDISC SEND Controlled Terminology, released by NCI on 2013-12-20	New CT package found in <code>newct.standardsubtypes</code> but not found in <code>oldct.standardsubtypes</code>

Compare Two Codelist Sources

Overview

To help you determine the differences between two codelists, SAS supplies the `cstutilcomparecodelists` macro.

`cstutilcomparecodelists` Macro

This example compares the SAS Clinical Standards Toolkit 1.6 codelist data set to the 1.5 codelist data set:

```
libname newct 'C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\control';
```

```
libname oldct 'C:\cstGlobalLibrary15\standards\cdisc-terminology-1.5\control';

%cstutilcomparecodelists(
    _cstFileType=DATASET,
    _cstBaseCT=oldct.cterms,
    _cstNewCT=newct.cterms,
    _cstCompareCL=Y,
    _cstCLVar=codelist,
    _cstCompareCLI=Y,
    _cstCLValueVar=cdisc_submission_value,
    _cstRptType=_CSTRESULTSDS
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

This display illustrates the data set created by the `cstutilcomparecodelists` macro:

Display 3.5 *Controlled Terminology Package Differences Reported by the `cstutilcomparecodelists` Macro*

	message	actual
2	Codelist found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN
3	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=ADOLESCENT (12-17 YEARS)
4	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=ADULT (18-65)
5	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=CHILDREN (2-11 YEARS)
6	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=ELDERLY (> 65)
7	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=IN UTERO
8	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=INFANT AND TODDLER (28 DAYS - 23 MONTHS)
9	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=NEWBORN (0-27 DAYS)
10	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=AGESPAN,Value=PRETERM NEWBORN INFANTS
11	Codelist found in oldct.ctems but not found in newct.ctems	Codelist=CCINVCTYP
12	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=CCINVCTYP,Value=CASUAL CONTACT
13	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=CCINVCTYP,Value=CONTACT WITH EXPOSURE DURING MEDICAL PROCEDURE
14	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=CCINVCTYP,Value=CONTACT WITH EXPOSURE IN CROWD SETTING
15	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=CCINVCTYP,Value=CONTACT WITH MEDICAL RISK FACTOR
16	Codelist value found in oldct.ctems but not found in newct.ctems	Codelist=CCINVCTYP,Value=HOUSEHOLD CONTACT
17	Codelist found in newct.ctems but not found in oldct.ctems	Codelist=CCINVTYP
18	Codelist value found in newct.ctems but not found in oldct.ctems	Codelist=CCINVTYP,Value=CASUAL CONTACT
19	Codelist value found in newct.ctems but not found in oldct.ctems	Codelist=CCINVTYP,Value=CONTACT WITH EXPOSURE DURING MEDICAL PROCEDURE

Rows 2 through 10 indicate that the AGESPAN codelist (including all values) has been removed from the CDISC-SDTM 201312 controlled terminology package. A comparison of rows 11 through 13 to rows 17 through 19 suggest that the CCINVCTYP codelist has been renamed CCINVTYP.

Compare Two Macro Autocall Libraries

Overview

To help you determine the differences in a library of SAS macros between two versions of the SAS Clinical Standards Toolkit, SAS supplies the `cstutilcompareautocallmacros` macro.

`cstutilcompareautocallmacros` Macro

The `cstutilcompareautocallmacros` macro creates a list of the contents of the directory for each of the two macro libraries.

The `cstutilcompareautocallmacros` macro compares the two lists and identifies any file that is contained in one library but not the other library. For each file that is common to both macro libraries, the macro indicates any changes in the macro signature.

This example compares a customized framework macro library to the framework macro library for the installed version of the SAS Clinical Standards Toolkit:

```
%cstutilcompareautocallmacros(  
  _cstBasePath=C:\Program Files\SASHome\SASFoundation\9.4\  
              cstframework_1.5customized\sasmacro,  
  _cstNewPath=C:\Program Files\SASHome\SASFoundation\9.4\cstframework  
              \sasmacro,  
  _cstRptType=_CSTRESULTSDS,  
  _cstOverwrite=N  
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

This display illustrates the data set created by the `cstutilcompareautocallmacros` macro (not all columns and rows are shown):

Display 3.6 Sample Results Data Set Created with the `cstutilcompareautocallmacros` Macro

	message	resultseverity	keyvalues
21	New macro (cstutilxmlvalidate.sas) not found in C:\Program Files\SASHome\SASFoundation\9.4\cstframework_1.5customized\sasmacro	Info	Macro name=cstutilxmlvalidate.sas
22	Old macro (csutilwriteresultsintro.sas) not found in C:\Program Files\SASHome\SASFoundation\9.4\cstframework\sasmacro	Info	Macro name=csutilwriteresultsintro.sas
23	New macro parameter(s) detected for: cst_createtablesfordatastandard.sas - see work.macrodiffs.	Info	Macro name=cst_createtablesfordatastandard.sas
24	New macro parameter(s) detected for: cstutil_deletedataset.sas - see work.macrodiffs.	Info	Macro name=cstutil_deletedataset.sas

Rows 21 and 22 indicate that there are differences in the files in the two macro libraries.

Rows 23 and 24 indicate that the macro signatures have changed. These changes are itemized in the WORK.MACRODIFFS data set:

Display 3.7 Macro Signature Differences Identified by the `cstutilcompareautocallmacros` Macro

name	parameter	baseSig	newSig
cst_createtablesfordatastandard.sas	_cstWhereClause	%macro cst_createtablesfordatastandard(_cstStandard=, _cstStandardVersion=, _cstOutputLibrary=, _cstResultsOVERRIDE=, _cstNumObs=0) / des = 'CST: Creates tables from registered referencemetadata.';	%macro cst_createtablesfordatastandard(_cstStandard=, _cstStandardVersion=, _cstOutputLibrary=, _cstWhereClause=, _cstResultsOVERRIDE=, _cstNumObs=0) / des = 'CST: Creates tables from registered referencemetadata.';
cstutil_deletedataset.sas	_cstLogging	%macro cstutil_deletedataset(_cstDataSetName=) / des='CST: Delete a SAS data set';	%macro cstutil_deletedataset(_cstDataSetName=, _cstLogging=0, _cstLoggingDS=) / des='CST: Delete a SAS data set';
cstutil_deletedataset.sas	_cstLoggingDS	%macro cstutil_deletedataset(_cstDataSetName=) / des='CST: Delete a SAS data set';	%macro cstutil_deletedataset(_cstDataSetName=, _cstLogging=0, _cstLoggingDS=) / des='CST: Delete a SAS data set';

Compare Two Property Files

Overview

A key element of the SAS Clinical Standards Toolkit is the set of global macro variables built into and used by the SAS Clinical Standards Toolkit. These global macro variables are primarily defined in property files.

For example, the *global standards library directory/standards/cdisc-sdtm-3.1.3-1.6/programs/initialize.properties* file includes the following name-value pairs that become global macro variables:

```
_cstStandard=CDISC-SDTM
_cstStandardVersion=3.1.3
_cstSubjectColumns=studyid usubjid
_cstTableMetadata=work._csttablemetadata
_cstColumnMetadata=work._cstcolumnmetadata
```

To help you determine whether there are changes in the global macro variables from one version of the SAS Clinical Standards Toolkit to another version, SAS supplies the `cstutilcompareproperties` macro.

cstutilcompareproperties Macro

The `cstutilcompareproperties` macro translates property files into SAS data sets for comparison.

This example compares the `initialize.properties` file for CDISC SDTM 3.1.3 to 3.2 in the SAS Clinical Standards Toolkit 1.6:

```
%cstutilcompareproperties(
    _cstBasePath=C:\cstGlobalLibrary\standards\
                cdisc-sdtm-3.1.3-1.6\programs\initialize.properties,
    _cstNewPath=C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\
                programs\initialize.properties
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

The example macro call outputs a message similar to this one in the SAS log file:

```
[CSTLOGMESSAGE] NOTE: The following properties differ between the two property files:
Property values different between BASE and NEW.
Name: _cstStandardVersion Base Value: 3.1.3 New Value: 3.2
```

Compare Two Folder Hierarchies

Overview

To help you determine whether two directories contain the same folder hierarchy (including subfolders), SAS supplies the `cstutilcomparefolderhierarchy` macro.

TIP For the best results, the subfolders in each directory must share a similar naming convention or structure, although this is not required.

`cstutilcomparefolderhierarchy` Macro

The `cstutilcomparefolderhierarchy` macro can compare these items:

- folders
The macro identifies the presence or absence of subfolders in the two folder hierarchies.
- files
The macro identifies the presence or absence of files in the two folder hierarchies.
- data sets
The macro identifies, based on the output of PROC COMPARE, the differences in the data sets that are common in the two folder hierarchies.

These comparisons can be run individually or simultaneously.

Example 1: Compare Global Standards Library Folder Hierarchies

This example compares the SAS Clinical Standards Toolkit 1.5 global standards library folder hierarchy (C:\cstGlobalLibrary15custom) to the 1.6 folder hierarchy (C:\cstGlobalLibrary) to determine which folders differ in the two versions:

```
%cstutilcomparefolderhierarchy(  
    _cstBaseFolder=C:\cstGlobalLibrary,  
    _cstBaseVersion=1.6,  
    _cstCompFolder=C:\cstGlobalLibrary15custom,  
    _cstCompVersion=1.5,  
    _cstRptDiff=work.foldercomp,  
    _cstRptDiffType=FOLDER,  
    _cstOverwrite=y,  
    _cstOutReportPath=C\  
    _cstOutReportFile=globallibrarydiff15.htm,  
);
```

Note: It does not matter which folder is `_cstBaseFolder` or `_cstCompFolder`.

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

This display illustrates part of the work.foldercomp data set created by the example:

Display 3.8 Part of the work.foldercomp Data Set Created by the Example

	BASE Folder name	COMP Folder name	Result
1	C:\cstGlobalLibrary\logs		Folder present in Base only
2	C:\cstGlobalLibrary\schemarepository\cdisc-definexml-2.0.0		Folder present in Base only
3	C:\cstGlobalLibrary\schemarepository\cdisc-odm-1.3.2		Folder present in Base only
4	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\control		Folder present in Base only
5	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\formats		Folder present in Base only
6	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\macros		Folder present in Base only
7	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\messages		Folder present in Base only
8	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\metadata		Folder present in Base only
9	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\programs		Folder present in Base only
10	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\stylesheet		Folder present in Base only
11	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\templates		Folder present in Base only
12	C:\cstGlobalLibrary\standards\cdisc-definexml-2.0.0-1.6\validation\control		Folder present in Base only
13	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\control		Folder present in Base only
14	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\macros		Folder present in Base only
15	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\messages		Folder present in Base only
16	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\metadata		Folder present in Base only
17	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\programs		Folder present in Base only
18	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\templates		Folder present in Base only
19	C:\cstGlobalLibrary\standards\cdisc-sdtm-3.2-1.6\validation\control		Folder present in Base only
20		C:\cstGlobalLibrary\15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104Extended\formats	Folder present in Comp only
21		C:\cstGlobalLibrary\15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104\formats	Folder present in Comp only
22	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-cdash\201312\formats		Folder present in Base only
23	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-qs\201312\formats		Folder present in Base only
24	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-qs\current\formats		Folder present in Base only
25		C:\cstGlobalLibrary\15custom\standards\cdisc-terminology-1.5\cdisc-sdtm\201104\formats	Folder present in Comp only
26	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-sdtm\201312\formats		Folder present in Base only
27		C:\cstGlobalLibrary\15custom\standards\cdisc-terminology-1.5\cdisc-send\201201\formats	Folder present in Comp only
28	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-send\201312\formats		Folder present in Base only
29		C:\cstGlobalLibrary\15custom\standards\cst-framework-1.5\customautocall	Folder present in Comp only
30	C:\cstGlobalLibrary\xsl-repository\DEFINE-XML\2.0.0\export		Folder present in Base only
31	C:\cstGlobalLibrary\xsl-repository\DEFINE-XML\2.0.0\export\config		Folder present in Base only
32	C:\cstGlobalLibrary\xsl-repository\DEFINE-XML\2.0.0\import		Folder present in Base only
33	C:\cstGlobalLibrary\xsl-repository\DEFINE-XML\2.0.0\import\config		Folder present in Base only

Note: Data shown might not be on your computer.

The first 19 rows identify the new folders in the base folder hierarchy. These folders represent new content in the SAS Clinical Standards Toolkit 1.6 that is not in 1.5.

Rows 20, 21, 25, 27, and 29 identify the folders in the comparison folder hierarchy (in the **COMP Folder name** column) that are not in the base folder hierarchy. (See the highlighted text in the previous display.)

For example, `C:\cstGlobalLibrary\15Custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104Extended\formats` (row 20) does not have a corresponding folder structure in 1.6 because this subfolder was created by the user to contain extended controlled terminology packages.

Likewise, `C:\cstGlobalLibrary\15Custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104\formats` (row 21) does not have a

corresponding folder structure in 1.6 because it was deprecated and replaced with a newer version (\201312).

Note: It is important to identify differences because you must decide whether to migrate these files into the SAS Clinical Standards Toolkit 1.6 or replace them with the newer 1.6 files. There are many factors to consider. For example, do the 1.5 controlled terminology packages contain extended codelists? Are the 1.5 controlled terminology packages customized for your organization or for a set of studies?

Example 2: Compare Global Standards Library Files

This example compares the SAS Clinical Standards Toolkit 1.5 global standards library folder hierarchy (C:\cstGlobalLibrary15Custom) to the 1.6 folder hierarchy (C:\cstGlobalLibrary) to determine which files differ in the two versions:

```
%cstutilcomparefolderhierarchy(  
    _cstBaseFolder=C:\cstGlobalLibrary,  
    _cstBaseVersion=1.6,  
    _cstCompFolder=C:\cstGlobalLibrary15Custom,  
    _cstCompVersion=1.5,  
    _cstRptDiff=work.filecomp,  
    _cstReportType=FILE,  
    _cstOverwrite=y  
);
```

Note: It does not matter which folder is `_cstBaseFolder` or `_cstCompFolder`.

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

This display illustrates rows 86 through 98 of the work.filecomp data set created by the example:

Display 3.9 Rows 86 through 98 of the work.filecomp Data Set Created by the Example

	BASE Folder name	COMP Folder name	Name of File	Result
86	C:\cstGlobalLibrary\standards\cdisc-sdm-3.2.1.6\validation\control		cported.xpt	File present in Base only
87	C:\cstGlobalLibrary\standards\cdisc-sdm-3.2.1.6\validation\control		validation_domainsbycheck.sas7bdat	File present in Base only
88	C:\cstGlobalLibrary\standards\cdisc-sdm-3.2.1.6\validation\control		validation_master.sas7bdat	File present in Base only
89	C:\cstGlobalLibrary\standards\cdisc-sdm-3.2.1.6\validation\control		validation_stdref.sas7bdat	File present in Base only
90	C:\cstGlobalLibrary\standards\cdisc-send-3.0-1.6\macros		sendutil_createformatsfromctdds.sas	File present in Base only
91		C:\cstGlobalLibrary15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104extended\formats	cported.xpt	File present in Comp only
92		C:\cstGlobalLibrary15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104extended\formats	cterext.sas7bcat	File present in Comp only
93		C:\cstGlobalLibrary15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104extended\formats	cterext.sas7bdat	File present in Comp only
94		C:\cstGlobalLibrary15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104\formats	cported.xpt	File present in Comp only
95		C:\cstGlobalLibrary15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104\formats	cterms.sas7bcat	File present in Comp only
96		C:\cstGlobalLibrary15custom\standards\cdisc-terminology-1.5\cdisc-cdash\201104\formats	cterms.sas7bdat	File present in Comp only
97	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-cdash\201312\formats		cported.xpt	File present in Base only
98	C:\cstGlobalLibrary\standards\cdisc-terminology-1.6\cdisc-cdash\201312\formats		cterms.sas7bcat	File present in Base only

Note: Data shown might not be on your computer.

Rows 86 through 90 identify the new files in the base folder hierarchy. These files represent new content in the SAS Clinical Standards Toolkit 1.6 that is not in 1.5.

Rows 91 through 96 identify the files from the two folders in the previous example. (See row 20 and 21 [Display 3.8 on page 31](#).) Six files are not in the base folder hierarchy (1.6).

Note: It is important to identify differences because you must decide whether to migrate these files into the SAS Clinical Standards Toolkit 1.6. If these files are unchanged and have been replaced by the files in 1.6, you probably do not need to migrate them. However, if you have customized these files for certain studies or organizational standards, you might need to migrate these files.

Rows 97 and 98 identify the new files in the base folder hierarchy. These files represent new content in the SAS Clinical Standards Toolkit 1.6 that is not in 1.5.

This display illustrates rows 1 through 6 of the work.filecomp data set created by the example:

Display 3.10 Rows 1 through 6 of the work.filecomp Data Set Created by the Example

	BASE Folder name	COMP Folder name	Name of File	Result
1	C:\cstGlobalLibrary\logs		transactionlog.sas7bdat	File present in Base only
2	C:\cstGlobalLibrary\schemarepository\cdisc-definexml-2.0.0		define-extension.xsd	File present in Base only
3	C:\cstGlobalLibrary\schemarepository\cdisc-definexml-2.0.0		define-ns.xsd	File present in Base only
4	C:\cstGlobalLibrary\schemarepository\cdisc-definexml-2.0.0		define2-0-0.xsd	File present in Base only
5		C:\cstGlobalLibrary15custom\schemarepository\cdisc-odm-1.2.1	ODM1-2-1-clean.xml	File present in Comp only
6	C:\cstGlobalLibrary\schemarepository\cdisc-odm-1.3.2		ODM1-3-2-foundation.xsd	File present in Base only

Rows 1 through 4 and 6 identify the new files in the base folder hierarchy. These files represent new content in the SAS Clinical Standards Toolkit 1.6 that is not in 1.5.

Note: You must decide whether these files must be migrated to 1.6.

Row 5 identifies the ODM1-2-2-clean.xml file that is not in 1.6.

Copy a Folder Hierarchy

Overview

To help you copy a folder, all of its subfolders, and all content, SAS supplies the `cstutilcopyfolderhierarchy` macro.

Although you can copy a folder hierarchy using operating system commands or tools, the `cstutilcopyfolderhierarchy` macro provides useful options, such as itemization of the folders and files copied and summarization of the actions performed.

`cstutilcopyfolderhierarchy` Macro

The `cstutilcopyfolderhierarchy` macro creates a list of the folders and files contained by a folder hierarchy. Two data sets are created: one that contains the folders and one that contains the files. You can use these data sets for other programs, such as renaming the folders and files.

The macro offers an option to copy only the source folder hierarchy without also copying the files contained in the source folders.

This example copies the source folder hierarchy `\\d78398\public\cstGlobalLibrary` to `C:\cstGlobalLibrary15`:

```
%cstutilcopyfolderhierarchy(  
    _cstSourceFolder=\\d78398\public\cstGlobalLibrary,  
    _cstNewFolder=C:\cstGlobalLibrary15,  
    _cstFolderDS=work.folders,  
    _cstFileDS=work.files,  
    _cstBuildFoldersOnly=N  
);
```

For more information about the macros and their parameters, see [Chapter 6, “Macro API,”](#) on page 53.

The example macro call outputs a message similar to this one in the SAS log file:

```
NOTE: [CSTLOGMESSAGE] 191 folders were created.
NOTE: [CSTLOGMESSAGE] 1049 files were copied.
```

This display illustrates the data set that contains the folders copied by the example:

Display 3.11 Sample Data Set That Contains the Folders Copied by the *cstutilcopyfolderhierarchy* Macro

	root
1	\\d78398\public\cstGlobalLibrary
2	\\d78398\public\cstGlobalLibrary\logs
3	\\d78398\public\cstGlobalLibrary\metadata
4	\\d78398\public\cstGlobalLibrary\schema-repository
5	\\d78398\public\cstGlobalLibrary\standards
6	\\d78398\public\cstGlobalLibrary\xsl-repository
7	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-crtdds-1.0.0
8	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-ct-1.0.0

This display illustrates the data set that contains the files copied by the example:

Display 3.12 Sample Data Set That Contains the Files Copied by the *cstutilcopyfolderhierarchy* Macro

	path	filename	fileextension
1	\\d78398\public\cstGlobalLibrary\logs	transactionlog.sas7bdat	sas7bdat
2	\\d78398\public\cstGlobalLibrary\metadata	availabletransforms.xml	xml
3	\\d78398\public\cstGlobalLibrary\metadata	standardlookup.sas7bdat	sas7bdat
4	\\d78398\public\cstGlobalLibrary\metadata	standards.sas7bdat	sas7bdat
5	\\d78398\public\cstGlobalLibrary\metadata	standardsasreferences.sas7bdat	sas7bdat
6	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-crtdds-1.0.0	define-extension.xsd	xsd
7	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-crtdds-1.0.0	define-ns.xsd	xsd
8	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-crtdds-1.0.0	define-1-0-0.xsd	xsd
9	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-ct-1.0.0	controlledterminology-extension.xsd	xsd
10	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-ct-1.0.0	controlledterminology-ns.xsd	xsd
11	\\d78398\public\cstGlobalLibrary\schema-repository\cdisc-ct-1.0.0	controlledterminology1-0-0.xsd	xsd

4

Tools to Migrate Source Metadata from CRT-DDS 1.0.0 to Define-XML 2.0.0

<i>Introduction</i>	37
<i>Overview</i>	38
<i>Study Source Metadata Migration Driver Programs</i>	39

Introduction

SAS supplies tools to support the migration of study source metadata that was created for a CRT-DDS 1.0.0 define.xml file to study source metadata to be used to create a Define-XML 2.0.0 define.xml file.

Note: Be aware that the study source metadata files created should not be considered ideal Define-XML 2.0.0 implementations for production use. Although the SAS tools provide a good starting point and a way to explore Define-XML 2.0.0 metadata, the files have limitations. In particular, the set of ValueList definitions and WHERE clauses are converted from value-level metadata definitions in CRT-DDS 1.0 that have limitations.

The tools are contained in the cstmigration.zip file. For more information about where this file is located and its structure, see [“Introduction” on page 14](#).

You can find an overview of the changes from the CRT-DDS 1.0.0 model to the Define-XML 2.0.0 model in the document *Define-XML v2 – What’s New*, which is located at <http://www.lexjansen.com/phuse/2013/cd/CD05.pdf>.

Overview

When the SAS Clinical Standards Toolkit creates a define.xml file for CRT-DDS 1.0.0 or Define-XML 2.0.0, this is the process:

- 1 Create data sets that contain the study source metadata.
- 2 Convert the study source metadata into the SAS representation of the CRT-DDS 1.0.0 or Define-XML 2.0.0 model.

- 3 Extend the SAS representation of the model, if needed.

For Define-XML 2.0.0, this step is not usually needed because in most cases the study source metadata can contain the complete metadata content for the creation of the Define-XML 2.0.0 file. The tools in this chapter assume that there are data sets with study source metadata for an ADaM or SDTM study to use to create a CRT-DDS 1.0.0 define.xml file.

Note: If the SAS representation of the model was extended to create a define.xml file for CRT-DDS 1.0.0, you must ensure that the study source metadata created by this step is added to the Define-XML 2.0.0 study source metadata. The conversion tool uses only the study source metadata data sets, not the SAS representation of the CRT-DDS 1.0.0 model.

- 4 Create a define.xml file from the SAS representation of the CRT-DDS 1.0.0 or Define-XML 2.0.0 model.

For CRT-DDS 1.0.0, the following study source metadata data sets are defined in the SAS Clinical Standards Toolkit 1.5:

- source_study
- source_tables
- source_columns
- source_values

- source_documents

For Define-XML 2.0.0, a new study source metadata data set (source_codelists) has been defined. The source_codelists data set was not part of the CRT-DDS 1.0.0 study source metadata data sets. The source_codelists data set contains all metadata needed to create the codelists in the define.xml file, including the external codelists (for example, MedDRA and WHO DRUG) and NCI metadata (for example, the so-called C-codes).

Study Source Metadata Migration Driver Programs

To provide an example of migrating CRT-DDS 1.0.0 source metadata to Define-XML 2.0.0 source metadata, SAS supplies the migrate_crtds_to_definexml_adam.sas and migrate_crtds_to_definexml_sdtm.sas driver programs.

Note: The driver programs for ADaM and SDTM are similar in structure, so this example addresses only the SDTM driver program.

After the driver program performs the initial setup (which you can review in the driver program), these librefs are defined:

```
%*****;
%* Define libnames for input *;
%*****;
%* Original CRT-DDS v1 source metadata for SDTM 3.1.3 in CST 1.5;
libname crtds "c:/cstSampleLibrary15/cdisc-sdtm-3.1.3-1.5/sascstdemodata/metadata";

%*****;
%* Define libnames for output *;
%*****;
%* Migrated Define-XML v2 source metadata;
libname defv2 "&_cstMigrationPath/definexml/sascstdemodata/cdisc-sdtm-3.1.3-1.6/metadata";

%*****;
%* Define formats *;
%*****;
%* SDTM Study formats in CST 1.5;
libname studyfmt "c:/cstSampleLibrary15/cdisc-sdtm-3.1.3-1.5/sascstdemodata/terminology/formats";
```

```

%* CDISC-NCI Terminology to be used in CST 1.6;
libname ncisdtm "c:/cstGlobalLibrary/standards/cdisc-terminology-1.6/cdisc-sdtm/201212/formats";

%* Formats to be used for SDTM;
options fmtsearch = (studyfmt.formats ncisdtm.terms);

```

Note: You might need to change these librefs.

Some of the CRT-DDS 1.0.0 metadata must be mapped to the values expected by Define-XML 2.0.0. It is likely that you must change the formats based on your specific data values. It is important to use the format names as specified because these formats are used by the conversion macros.

```

%*****;
%* Create some formats for mapping *;
%*****;
proc format;
  value $_cststd
    /* Maps from CRT-DDS values to required Define-XML v2 values */
    "CDISC SDTM"="SDTM-IG"
    "CDISC SEND"="SEND-IG"
    "CDISC ADAM"="ADAM-IG"
  ;
  value $_cstdom
    /* Map to ItemGroup/@Domain attribute */
    "QSCG" = "QS"
    "QSCS" = "QS"
    "QSMM" = "QS"
  ;
  value $_cstdomd
    /* Map to ItemGroup/Alias[@Context='DomainDescription']/@Name attribute */
    "QSCG" = "Questionnaires"
    "QSCS" = "Questionnaires"
    "QSMM" = "Questionnaires"
  ;
  value $_cstcls
    /* Maps from CRT-DDS values to required Define-XML v2 values */
    "SPECIAL PURPOSE DOMAINS" = "SPECIAL PURPOSE"
    "SPECIAL PURPOSE DATASETS" = "SPECIAL PURPOSE"
    "FINDINGS ABOUT" = "FINDINGS"
    "ADSL" = "SUBJECT LEVEL ANALYSIS DATASET"
    "ADAE" = "ADAM OTHER"
    "BDS" = "BASIC DATA STRUCTURE"
  ;
  value $_cstvlm
    /* For SDTM maps to variables that are being described by Value Level Metadata */
    "EG.EGTESTCD" = "EGORRES"

```



```

"IE.IETESTCD" = "IEORRES"
"TI.IETESTCD" = "IECAT"
"LB.LBTESTCD" = "LBORRES"
"PE.PETESTCD" = "PEORRES"
"SC.SCTESTCD" = "SCORRES"
"VS.VSTESTCD" = "VSORRES"
"SUPPAE.QNAM" = "QVAL"
;
run;

```

Now, the metadata is converted.

```

%*****;
%* Define the studyversion macro variable. *;
%* This will become the MetaDataVersion/@OID attribute *;
%* In CRT-DDS this was the source_study.definedocumentname column *;
%* Also define the SASRef macro variable to use for the SASRef column in the *;
%* source_XXX data sets. *;
%*****;
proc sql noprint;
  select definedocumentname, SASRef into :studyversion, :SASRef
  from crtdds.source_study;
quit;

%*****;
%* Migrate source tables *;
%*****;
%cstutilmigratecrtdds2define(
  _cstSrcLib=crtdds, _cstSrcDS=source_study, _cstTrgDS=defv2.source_study,
  _cstStudyVersion=&studyversion, _cstStandard=&_cstTrgStandard);
%cstutilmigratecrtdds2define(_cstSrcLib=crtdds, _cstSrcDS=source_tables,
  _cstTrgDS=defv2.source_tables, _cstStudyVersion=&studyversion,
  _cstStandard=&_cstTrgStandard);
%cstutilmigratecrtdds2define(_cstSrcLib=crtdds, _cstSrcDS=source_columns,
  _cstTrgDS=defv2.source_columns, _cstStudyVersion=&studyversion,
  _cstStandard=&_cstTrgStandard);
%cstutilmigratecrtdds2define(_cstSrcLib=crtdds, _cstSrcDS=source_values,
  _cstTrgDS=defv2.source_values, _cstStudyVersion=&studyversion,
  _cstStandard=&_cstTrgStandard);
%cstutilmigratecrtdds2define(_cstSrcLib=crtdds, _cstSrcDS=source_documents,
  _cstTrgDS=defv2.source_documents, _cstStudyVersion=&studyversion,
  _cstStandard=&_cstTrgStandard
);

```

Now, the source_codelists table is created. This is a separate process because this table was not in the CRT-DDS 1.0.0 source metadata.

```

%*****;

```

```

%* Create source_codelists                                     *;
%*****;
%* Get formats ;
%cstutilgetncimetadata(
    _cstFormatCatalogs=,
    _cstNCICTerms=ncisdtn.cterms,
    _cstLang=en,
    _cstStudyVersion=&studyversion,
    _cstStandard=&_cstTrgStandard,
    _cstStandardVersion=&_cstTrgStandardVersion,
    _cstFmtDS=work._cstformats,
    _cstSASRef=&SASRef,
    _cstReturn=_cst_rc,
    _cstReturnMsg=_cst_rcmsg
);

%* Create a data set with all applicable formats. ;
data work.cl_column_value(keep=xmlcodelist);
    set crtdds.source_columns crtdds.source_values;
    xmlcodelist=upcase(xmlcodelist);
    if xmlcodelist ne '';
run;

proc sort data=work.cl_column_value nodupkey;
    by xmlcodelist;
run;

%* Only keep applicable formats. ;
proc sql;
    create table defv2.source_codelists
    as select
        nci.*
    from
        work._cstformats nci, work.cl_column_value cv
    where (upcase(compress(nci.sasformatname, '$')) =
           upcase(compress(cv.xmlcodelist, '$')))
    ;
quit;

```

Note: The `_cstFormatCatalogs` parameter is blank. This indicates that the format catalogs that define the codelists to include in the `source_codelists` table are taken from the value specified for the `FMTSEARCH` option.

Finally, the metadata for external controlled terminology is added to the `source_codelists` data set.

```

%*****;

```

```

%* Updates for External Controlled Terminology *;
%*****;
proc sql;
insert into defv2.source_codelists
  (sasref, codelist, codelistname, codelistdatatype, sasformatname, dictionary, version,
  studyversion, standard, standardversion)
  values (&SASRef, "AEDICT_F", "Adverse Event Dictionary", "text", "$AEDICT", "MEDDRA",
  "8.0", "&studyversion", "&_cstTrgStandard", "&_cstTrgStandardVersion")
  ;
quit;

data defv2.source_columns;
set defv2.source_columns;
if table="ADAE" and column in ("AELLT" "AELLTCD" "AEDECOD" "AEPTCD" "AEHLT" "AEHLTCD"
  "AEHLGT" "AEBODSYS" "AESOC")
then xmlcodelist="AEDICT";
run;

```




Frequently Asked Questions

<i>Installation Issues</i>	46
Backup Copies	46
Overwrite Previous Versions of the SAS Clinical Standards Toolkit	46
Install the SAS Clinical Standards Toolkit 1.6 in Conjunction with 1.5	46
Upgrade Prior Versions of the SAS Clinical Standards Toolkit	47
<i>Migration Process</i>	47
Order of Migration	47
<i>Standards Management</i>	48
Unused Standards	48
<i>Library Issues</i>	48
Delete Sample Study Library	48
Maintain Metadata Attributes	49
cstGlobalLibrary Standards Folder	49
Update Customized Standard Metadata across Global Standards Libraries	50
<i>Other Issues</i>	50
Driver Programs	50

Installation Issues

Backup Copies

Is it necessary to back up my copy of the SAS Clinical Standards Toolkit 1.5 before I install 1.6? If I have customized my SAS Clinical Standards Toolkit, what must I change in the configuration file?

If you have customized any of the three locations that SAS Clinical Standards Toolkit writes to during the installation process, you must back up the files. (For more information about these locations, see [“Installation Locations” on page 1.](#))

A new installation of SAS does not typically replace the old configuration file. If you have concerns about modifications that you made to the configuration file, back up the file to ensure that you can restore it.

Overwrite Previous Versions of the SAS Clinical Standards Toolkit

Will an installation of the SAS Clinical Standards Toolkit 1.6 overwrite 1.5?

Only partially. As described in the [“Installation Locations” on page 1](#), the SAS Clinical Standards Toolkit places files in three locations. Only files in the !sasroot location can be overwritten.

For a full description of an installation process that requires you to install the global standards library and sample study library to other locations, see the [“SAS Deployment Wizard” on page 5](#).

Install the SAS Clinical Standards Toolkit 1.6 in Conjunction with 1.5

Can I install the SAS Clinical Standards Toolkit 1.6 in conjunction with 1.5?

Yes. For more information, see “Upgrade to SAS Clinical Standards Toolkit 1.6 By Installing SAS 9.4 in Conjunction with SAS 9.3 (TS1M2)” on page 9.

Upgrade Prior Versions of the SAS Clinical Standards Toolkit

I am running a version of the SAS Clinical Standards Toolkit prior to 1.5. Do I need to separately upgrade to each version to get up-to-date to 1.6?

No. If you did not customize previous versions of the SAS Clinical Standards Toolkit, and you do not require access to any functionality that has been deprecated (such as framework autocall macros and older controlled terminology packages), you can use the full functionality provided by the SAS Clinical Standards Toolkit 1.6.

Note: Deprecations are generally noted in the “What’s New” section of the *SAS Clinical Standards Toolkit: User's Guide* for each version.

If you customized previous versions of the SAS Clinical Standards Toolkit, run the migration tools to identify file differences that you might need to address.

Migration Process

Order of Migration

Is there a suggested order to migrate data? Is all or most of the work centered on the SAS Clinical Standards Toolkit global standards library?

Yes, this is the suggested order to migrate data:

- 1 Migrate user customizations that affect the global standards library. New standards or customized versions of standards are registered in the global standards library. Copy these user customizations into the SAS Clinical Standards Toolkit 1.6 global standards library folder hierarchy and register them to 1.6.
- 2 If recommended best practices were not followed, migrate user customizations to the !sasroot autocall library.

- 3 Migrate changes to the sample study library. These changes are least likely to be impacted unless you have chosen to build a sample study for your customized standards or you have modified the driver programs. A typical modification to a driver program is to reference a new sample study root path location.

Standards Management

Unused Standards

Can I delete from the global standards library any unused standards that are supplied with the SAS Clinical Standards Toolkit? If I delete the folders, does this affect any of my metadata files in the framework?

Run the `cst_unregisterstandard` macro for each unwanted standard and `standardversion`. This macro does not remove the associated folder hierarchies from the global standards library and the sample study library. You can delete the folder hierarchies later.

Some framework sample driver programs that support internal validation include references to all standards and standard versions that are supplied by SAS. These references typically appear in `WHERE` clauses. These `WHERE` clauses return fewer records (standards or standard versions) to validate.

Library Issues

Delete Sample Study Library

Do I need the SAS Clinical Standards Toolkit 1.6 `cstSampleLibrary` or can I delete it?

The sample study library provides sample study implementations (including metadata, data, and code) to demonstrate the SAS Clinical Standards Toolkit functionality for various standards that are supplied by SAS. Without the sample study implementations,

all sample processes and internal validation checks fail. Furthermore, product documentation assumes its presence.

However, the sample study library is not necessary in a mature SAS Clinical Standards Toolkit environment in which users have their own processes that point to their own study data. The `studylibraryrootpath` column in the *global standards library directory/metadata/standards* data set (if used) is probably set to some study repository root path location outside the sample study library folder hierarchy.

Maintain Metadata Attributes

I have used the default global standards library for metadata attributes in all of my studies. For example, I populated my `source_*` data sets with default information. What must I do to maintain the changes to the source metadata that I have?

Most changes in the reference metadata representations of each CDISC standard reflect updates made by CDISC. If you do not already have a process in place to identify and review these changes with each CDISC standard version, you can compare the SAS Clinical Standards Toolkit 1.5 and 1.6 reference metadata for each standard or standardversion of interest. After the changes are identified and reviewed, update any `source_*` data sets to reflect the changes.

A suite of tools to identify differences in folder hierarchies and files is described in [“Tools to Identify Differences between Versions” on page 14](#).

cstGlobalLibrary Standards Folder

I notice in the `cstGlobalLibrary` standards folder that the SAS Clinical Standards Toolkit version is appended to the folder. I have many changes to the SDTM 3.1.3 standard folder in my SAS Clinical Standards Toolkit 1.5. Can I continue to use that folder and ignore the SDTM 3.1.3 standard folder in 1.6 because I want to preserve my changes?

Yes. Consider copying and registering the SDTM 3.1.3 standard folder in your SAS Clinical Standards Toolkit 1.5 as a custom version of SDTM 3.1.3 in 1.6. No structural changes to metadata between SAS Clinical Standards Toolkit 1.5 and 1.6 should interfere with this strategy.

Update Customized Standard Metadata across Global Standards Libraries

I work at a CRO. We have more than 30 global standards libraries that contain customized standard metadata for customers. What is the easiest way for me to update any information that is specific to the SAS Clinical Standards Toolkit 1.6 across these global standards libraries? Is it a manual process? Are there any available tools that can aid in this process?

It is not possible to provide definitive migration guidance without knowing what metadata has been customized and whether these customizations are in the SAS deployment locations for the SAS Clinical Standards Toolkit.

However, your migration process will be simpler if you have standardized your customizations across global standards libraries and if your customizations have not been made to the SAS deployment locations for the SAS Clinical Standards Toolkit.

Other Issues

Driver Programs

Will the driver programs that I created for my studies in the SAS Clinical Standards Toolkit 1.5 run without any changes in 1.6? If not, what do I need to look for?

Although it is not possible to answer this question with confidence without knowing the details of your driver program customizations, SAS expects most driver programs to work in both version 1.5 and 1.6 with no or minimal modification.

These coding changes might be required, depending on your coding style and macro use:

- If you have hardcoded the SAS Clinical Standards Toolkit version as 1.5 or if you have hardcoded the location of the sample study library, consider replacing the hardcoded locations with code such as this:

```
%cstutil_setcstsroot;
```

```
data _null_;  
  call symput('studyRootPath',cats("&_cstSRoot",  
    "/cdisc-sdtm-&_cstStandardVersion-&_cstVersion/sascstdemodata"));  
  call symput('studyOutputPath',cats("&_cstSRoot",  
    "/cdisc-sdtm-&_cstStandardVersion-&_cstVersion/sascstdemodata"));  
run;
```

- If your driver program references deprecated macros or deprecated controlled terminology packages, you must reference alternatives.
- If your SASReferences file includes any paths that are no longer valid, correct the paths.

6

Macro API

<i>Overview</i>	53
-----------------------	----

Overview

Detailed documentation of the macros is available in the file *root of unzipped folder hierarchy/macro-api-doc/index.html*.

