

# **SAS<sup>®</sup> 9.3 CDISC Procedure User's Guide**



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *SAS® 9.3 CDISC Procedure: User's Guide*. Cary, NC: SAS Institute Inc.

**SAS® 9.3 CDISC Procedure: User's Guide**

Copyright © 2011, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

**For a hardcopy 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 Restricted Rights Notice:** Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227–19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st electronic book, July 2011

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at

[support.sas.com/publishing](http://support.sas.com/publishing) 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 registered trademarks or trademarks of their respective companies.

---

# Contents

*Recommended Reading* . . . . . v

## PART 1 Concepts 1

<b>Chapter 1 • Introduction to the CDISC Procedure</b> . . . . .	<b>3</b>
What Is CDISC? . . . . .	3
What Is the CDISC Procedure? . . . . .	3
Overview of PROC CDISC Syntax . . . . .	4
Accessibility Features of the CDISC Procedure . . . . .	4
<b>Chapter 2 • Importing and Exporting a CDISC ODM XML Document</b> . . . . .	<b>5</b>
Introduction to CDISC ODM . . . . .	5
CDISC ODM KeySet Members . . . . .	6
Specifying CDISC ODM Metadata Attributes . . . . .	9
<b>Chapter 3 • Validating a CDISC SDTM SAS Data Set</b> . . . . .	<b>13</b>
Introduction to CDISC SDTM . . . . .	13
Validating a CDISC SDTM SAS Data Set . . . . .	13
CDISC Procedure Capabilities for Validating a CDISC SDTM SAS Data Set . . . . .	14

## PART 2 Usage 15

<b>Chapter 4 • CDISC Procedure Examples for CDISC ODM</b> . . . . .	<b>17</b>
Importing a CDISC ODM XML Document Using Default KeySet Processing . . . . .	18
Importing a CDISC ODM XML Document Specifying KeySet Processing Options . . . . .	20
Importing a CDISC ODM XML Document Using a Language Identifier . . . . .	22
Importing a CDISC ODM XML Document with OrderNumber Attributes . . . . .	28
Exporting a CDISC ODM XML Document with Metadata	
Attributes in Statement Syntax . . . . .	29
Exporting a CDISC ODM XML Document with Metadata	
Attributes in SAS Data Sets . . . . .	37
Exporting a CDISC ODM XML Document with Required and Optional Statements . . . . .	42
Describing a CDISC ODM SAS Data Set with the CONTENTS Statement . . . . .	45
Listing a Directory with the DATASETS Statement . . . . .	50
<b>Chapter 5 • CDISC Procedure Examples for CDISC SDTM</b> . . . . .	<b>53</b>
Validating CDISC SDTM Data in a SAS Data Set . . . . .	53
Validating CDISC SDTM Data in an Oracle Table . . . . .	54

## PART 3 Procedure Reference 57

<b>Chapter 6 • CDISC ODM Procedure</b> . . . . .	<b>59</b>
--	-----------

Overview: CDISC ODM Procedure .....	59
Syntax: CDISC ODM Procedure .....	60
<b>Chapter 7 • CDISC SDTM Procedure .....</b>	<b>83</b>
Overview: CDISC SDTM Procedure .....	83
Syntax: CDISC SDTM Procedure .....	83
<b>PART 4 Appendixes</b>	<b>87</b>
<b>Appendix 1 • Sample CDISC ODM XML Document .....</b>	<b>89</b>
<b>Index .....</b>	<b>97</b>

# Recommended Reading

---

- *SAS XML LIBNAME Engine: User's Guide*
- SAS Companion that is specific to your operating environment
- Base SAS focus area at [support.sas.com/rnd/base](http://support.sas.com/rnd/base)
- For information about XML (Extensible Markup Language), see the Web site [www.w3.org/XML](http://www.w3.org/XML)

For a complete list of SAS publications, go to [support.sas.com/bookstore](http://support.sas.com/bookstore). If you have questions about which titles you need, please contact a SAS Publishing Sales Representative:

SAS Publishing Sales  
SAS Campus Drive  
Cary, NC 27513-2414  
Phone: 1-800-727-3228  
Fax: 1-919-677-8166  
E-mail: [sasbook@sas.com](mailto:sasbook@sas.com)  
Web address: [support.sas.com/bookstore](http://support.sas.com/bookstore)



## Part 1

---

# Concepts

<i>Chapter 1</i>	
<b>Introduction to the CDISC Procedure</b> .....	<b>3</b>
<i>Chapter 2</i>	
<b>Importing and Exporting a CDISC ODM XML Document</b> .....	<b>5</b>
<i>Chapter 3</i>	
<b>Validating a CDISC SDTM SAS Data Set</b> .....	<b>13</b>





## Chapter 1

# Introduction to the CDISC Procedure

---

What Is CDISC? .....	3
What Is the CDISC Procedure? .....	3
Overview of PROC CDISC Syntax .....	4
Accessibility Features of the CDISC Procedure .....	4

---

## What Is CDISC?

CDISC (Clinical Data Interchange Standards Consortium) is an organization that develops industry standards. The standards support the electronic acquisition, exchange, and archival of clinical trials data and metadata for medical and biopharmaceutical product development. CDISC defines several data models for the interoperability of clinical data exchange.

The mission of CDISC is to develop and support global, platform-independent data standards that enable information system interoperability to improve medical research and related areas of health care.

For more information about CDISC, see the Web site at [www.cdisc.org](http://www.cdisc.org).

*Note:* The CDISC mission statement above is cited from the CDISC Web site. In addition, some explanations in the CDISC procedure documentation are from the CDISC standards descriptions.

---

## What Is the CDISC Procedure?

The CDISC procedure provides functionality that is based on specific CDISC models. The procedure currently supports the following CDISC models and functionality:

### CDISC ODM version 1.2

PROC CDISC provides the ability to import and export XML documents that conform to CDISC ODM version 1.2.

- Importing is the process of reading an external XML document as a SAS data set.
- Exporting is the process of writing a SAS data set to an output XML document that conforms to a CDISC model.

*Note:* For CDISC ODM support after version 1.2, see SAS Clinical Data Integration.

CDISC SDTM version 3.1

PROC CDISC performs data content validation on a SAS data set that conforms to CDISC SDTM version 3.1. PROC CDISC validates the SAS data set against domain definitions that are provided by CDISC SDTM.

*Note:* For CDISC SDTM support after version 3.1, see the SAS Clinical Standards Toolkit and SAS Clinical Data Integration.

**TIP** A SAS data set is any file that is accessed by SAS, such as a SAS data file or a file that points to data from other sources, such as a DBMS table that is accessed with a SAS/ACCESS engine.

---

## Overview of PROC CDISC Syntax

The syntax for PROC CDISC depends on the following:

- The CDISC model determines the set of associated statements. That is, CDISC ODM has a set of associated statements, and CDISC SDTM has a set of associated statements.
- For CDISC ODM, whether you want to import or export a file determines what statements are required or optional. In addition, each statement supports both required and optional syntax, depending on the process.

The following PROC CDISC code shows the syntax that is required to import an XML document that conforms to CDISC ODM version 1.2:

```
proc cdisc model=odm read=xmlinp;  
  odm odmversion="1.2";  
  clinicaldata out=results.ae sasdatasetname="AE";  
run;
```

The following PROC CDISC code shows the syntax that is required to validate a SAS data set that conforms to CDISC SDTM version 3.1:

```
proc cdisc model=sdm;  
  sdm sdmversion="3.1";  
  domaindata data=results.ae domain=ae catagory=events;  
run;
```

---

## Accessibility Features of the CDISC Procedure

The CDISC procedure is a command-based product. For this release, no features were added to address accessibility, but the product might very well be compliant to accessibility standards because it does not have a graphical user interface, and all of its features are available to anyone who can type or otherwise produce a command. If you have specific questions about the accessibility of SAS products, send them to [accessibility@sas.com](mailto:accessibility@sas.com) or call SAS Technical Support.

## Chapter 2

# Importing and Exporting a CDISC ODM XML Document

---

<b>Introduction to CDISC ODM</b> .....	<b>5</b>
What Is CDISC ODM? .....	5
CDISC ODM Basics .....	5
<b>CDISC ODM KeySet Members</b> .....	<b>6</b>
What Is a CDISC ODM KeySet? .....	6
Converting CDISC ODM KeySet Members .....	7
Processing CDISC ODM KeySet Members .....	9
<b>Specifying CDISC ODM Metadata Attributes</b> .....	<b>9</b>
Introduction to CDISC ODM Metadata Attributes .....	9
Specifying CDISC ODM Metadata Attributes When Importing .....	10
Specifying CDISC ODM Metadata Attributes When Exporting .....	11

---

## Introduction to CDISC ODM

### *What Is CDISC ODM?*

CDISC ODM is a vendor-neutral, platform-independent format. CDISC ODM supports the electronic acquisition, exchange, and archival of clinical trials data for the medical and biopharmaceutical industries.

PROC CDISC supports CDISC ODM version 1.2.

### *CDISC ODM Basics*

CDISC ODM defines the following entities to represent clinical trials (study) data:

item

describes an individual study item, such as a single blood pressure reading.

item group

describes a closely related set of study items that are usually analyzed together.

form

describes logically or temporally related information. Forms can contain multiple item groups.

study event

describes a data collection event such as a patient visit. A series of forms might be collected as part of a study event. A study event is associated with a specific patient in the study.

CDISC ODM defines the following metadata to describe the types of study events, forms, item groups, and items that are allowed in the study:

StudyEventDef

describes a particular instance of a study event.

FormDef

describes a particular instance of a form.

ItemDef

describes a particular instance of an item.

ItemGroupDef

describes a particular instance of an item group.

CodeList

defines a discrete set of permitted values for an item or a more human-readable description of an encoded value.

Metadata contains internal and external attributes that identify the entity for which information is provided. The internal attributes designate entities within the model and allow cross-references to be defined between entities, both within and between CDISC ODM XML documents. Internal attributes include an object instance identifier (OID), a subject key, and repeat keys.

- An OID uniquely identifies each entity. For example, a StudyOID is assigned to uniquely identify each study, a StudyEventOID is assigned to uniquely identify each StudyEventDef within a study, a FormOID is assigned to uniquely identify each form used in a StudyEventDef, and so on.
- The subject key identifies a subject within a study.
- The repeat key identifies an entity as one of a series. For example, there can be several study events of a particular type for a particular subject. The repeat key temporarily relates the distinct study events to each other.

The external attributes are used by clinical personnel to specify information that is unique to the entity. These attributes include subject randomization codes, site codes, and so on. Within CDISC ODM, the attributes are treated as part of the clinical trials data.

The set of attributes that are required to reference a single entity is referred to as a KeySet. For more information about the CDISC ODM KeySet, see [“CDISC ODM KeySet Members” on page 6](#).

---

## CDISC ODM KeySet Members

### *What Is a CDISC ODM KeySet?*

In a CDISC ODM XML document, a clinical trials data KeySet references an entity, such as a study, a subject, a study event, and so on. The following is an excerpt from a sample XML document that shows a fully populated KeySet:

```

<ClinicalData StudyOID="123-456-789" MetaDataVersionOID="v1.1.0">
  <SubjectData SubjectKey="001">
    <StudyEventData StudyEventOID="SE.VISIT1" StudyEventRepeatKey="1">
      <FormData FormOID="FORM.AE" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="1"
          TransactionType="Insert">

```

## Converting CDISC ODM KeySet Members

PROC CDISC imports and exports CDISC ODM KeySet members as follows:

- When importing, KeySet members (KeySet attributes) that are in the input XML document are converted to SAS variables and values in the output SAS data set.
- When exporting, KeySet members (SAS variables and values) that are in the input SAS data set are converted to KeySet attributes in the output XML document.

The following tables list the results of importing and exporting KeySet members:

**Table 2.1** Importing KeySet Members from a CDISC ODM XML Document

KeySet Member	Description	Resulting SAS Variable Name *
StudyOID=	Specifies a unique identifier for the study.	__STUDYOID
MetaDataVersionOID=	Specifies the metadata version that is used by the study.	__METADATAVERSIONOID
SubjectKey=	Specifies a subject within the study.	__SUBJECTKEY
StudyEventOID=	Specifies a StudyEventDef in the study.	__STUDYEVENTOID
StudyEventRepeatKey=	Specifies a study event repeat key.	__STUDYEVENTREPEATKEY
FormOID=	Specifies a FormDef in the study.	__FORMOID
FormRepeatKey=	Distinguishes between repeats of the same type of form in a single study event.	__FORMREPEATKEY
ItemGroupOID=	Specifies an ItemGroup in the study.	__ITEMGROUPOID
ItemGroupRepeatKey=	Specifies an ItemGroup repeat key.	__ITEMGROUPREPEATKEY

KeySet Member	Description	Resulting SAS Variable Name *
TransactionType=	Specifies the transaction type, which can be Insert, Update, Remove, Upsert, or Context.	__TRANSACTIONTYPE

\* The first two characters in resulting SAS variable names are two underscores to ensure that the name is unique.

**Table 2.2** Exporting KeySet Members from a SAS Data Set

KeySet Member	Description	Resulting KeySet Attributes	Created By
__STUDYOID	Specifies a unique identifier for the study.	StudyOID=	STUDY statement
__METADATAVERSIONOID	Specifies the metadata version that is used by the study.	MetaDataVersionOID=	METADATAVERSION statement
__SUBJECTKEY	Specifies a subject in the study.	SubjectKey=	Read from the exported SAS data set
__STUDYEVENTOID	Specifies a StudyEventDef in the study.	StudyEventOID=	STUDY statement
__STUDYEVENTREPEATKEY	Specifies a study event repeat key.	StudyEventRepeatKey=	Automatically generated
__FORMOID	Specifies a FormDef in the study.	FormOID=	Automatically generated
__FORMREPEATKEY	Distinguishes between repeats of the same type of form in a single study event.	FormRepeatKey=	Automatically generated
__ITEMGROUPOID	Specifies an ItemGroup in the study.	ItemGroupOID=	Automatically generated
__ITEMGROUPREPEATKEY	Specifies an ItemGroup repeat key.	ItemGroupRepeatKey=	Automatically generated

KeySet Member	Description	Resulting KeySet Attributes	Created By
__TRANSACTIONTYPE	Specifies the transaction type, which can be Insert, Update, Remove, Upsert, or Context.	TransactionType=	INSERT is always used

## Processing CDISC ODM KeySet Members

PROC CDISC provides processing options that determine how KeySet members are imported or exported. The following processing options are specified in the ODM statement:

LONGNAMES=NO | YES

when importing or exporting, determines the sources of captured SAS name parameters and controls the maximum length of SAS names that is valid.

ODMMAXIMUMOIDLENGTH=*number*

when importing, specifies a character length for CDISC ODM KeySet members.

ODMMINIMUMKEYSET=NO|YES

when importing or exporting, specifies whether to limit CDISC ODM KeySet members that are in the study data.

For more information about processing options, see the [ODM Statement on page 63](#).

For examples of how the KeySet processing options affect the data in an import operation, see “[Importing a CDISC ODM XML Document Using Default KeySet Processing](#)” on page 18 and “[Importing a CDISC ODM XML Document Specifying KeySet Processing Options](#)” on page 20.

---

## Specifying CDISC ODM Metadata Attributes

### Introduction to CDISC ODM Metadata Attributes

Several CDISC procedure statements enable you to specify metadata attributes either directly in the CDISC procedure statement or stored in a SAS data set that you reference in the DATA= argument.

The DATA= argument enables you to use the same execution code for all operations. It also enables you to change the metadata and data content by redirecting the LIBNAME statement specifications to different locations, perhaps on a study basis.

Here is an example of an ODM statement that specifies metadata attributes as part of the statement syntax:

```
odm odmversion="1.2"
  fileoid="000-00-0000"
  filetype=SNAPSHOT
  description="Adverse events from the CTChicago file";
```

The following example references those same metadata attributes stored in a SAS data set:

```
odm data=current.odm;
```

The SAS DATA step code shows how CURRENT.ODM, which contains the metadata attributes, is created:

```
data current.odm;
  odmversion="1.2";
  fileoid="000-00-0000";
  filetype="SNAPSHOT";
  description="Adverse events from the CTChicago file";
run;
```

The following example illustrates how the same PROC CDISC code can be used to export data for many studies:

```
libname metadata 'c:\your-meta-library';

libname clindata 'c:\your-data-library';

filename xmlout 'c:\your-output\ae.xml';

proc cdisc model=odm write=xmlout;
  odm data=metadata.odm;
  study data=metadata.study;
  globalvariables data=metadata.globals;
  basicdefinitions data=metadata.basic;
  metadataversion data=metadata.metadata;
  presentation data=metadata.present;
  user data=metadata.users;
  location data=metadata.location;
  signature data=metadata.signature;
  clinicaldata data=clindata.ae;
run;
```

### **Specifying CDISC ODM Metadata Attributes When Importing**

When importing a CDISC ODM XML document, you can specify the metadata attribute for the version number either directly in the ODM statement or stored in a SAS data set that you reference in the DATA= argument.

The following example imports data from a CDISC ODM ItemGroupDef element that has a SASDatasetName= attribute value of AE. All KeySet members are written to the output SAS data set named MY.AE. A maximum OID length of 16 characters is allocated for each KeySet member. The metadata attribute for the ODM statement is specified in the statement.

```
filename xmlinp 'CDISC-ODM-XML-document';

proc cdisc model=odm read=xmlinp formatactive=yes formatnoreplace=no;
  odm odmversion="1.2" odmmaximumoidlength=16 odmminimumkeyset=no;
  clinicaldata out=my.ae sasdatasetname="AE";
run;
```



## Specifying CDISC ODM Metadata Attributes When Exporting

When exporting a CDISC ODM XML document, follow these conventions:

- You can specify metadata attributes for the required statements ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION either directly in the statement or stored in a SAS data set that you reference in the DATA= argument.

*Note:* You cannot store some metadata attributes in a SAS data set, and also specify other metadata attributes in the statement. If you use a SAS data set, you must store all metadata attributes in the SAS data set.

- Optional metadata attributes for the CLINICALDATA statement must be specified as part of the statement syntax. For example, in the following code, the SAS data set MY.AE contains the KeySet members and clinical data content. DOMAIN=, NAME=, and COMMENT= are optional export metadata attributes.

```
clinicaldata data=my.AE
  domain="AE"
  name="Adverse Events"
  comment="All adverse events in this trial";
```

- Metadata attributes for the optional statements BASICDEFINITIONS, PRESENTATION, USER, LOCATION, and SIGNATURE must be submitted in a SAS data set that you reference in the DATA= argument.



## Chapter 3

# Validating a CDISC SDTM SAS Data Set

---

<b>Introduction to CDISC SDTM</b> .....	<b>13</b>
<b>Validating a CDISC SDTM SAS Data Set</b> .....	<b>13</b>
<b>CDISC Procedure Capabilities for Validating a CDISC SDTM SAS Data Set</b> . . .	<b>14</b>
Domain Content Validation .....	14
Unsupported Operations .....	14

---

## Introduction to CDISC SDTM

CDISC SDTM defines a standard structure for study data tabulation data sets that are submitted as part of a product application to a regulatory authority such as the U.S. Food and Drug Administration (FDA). CDISC SDTM was prepared by the CDISC Submissions Data Standards (SDS) team to guide the organization, structure, and format of the data sets. The data sets are one of four ways to represent the human subject Case Report Tabulation (CRT) and equivalent animal data submitted to the FDA.

CDISC SDTM includes several defined domains that are grouped within broad categories. The model provides the ability to create custom-defined domains with sets of standard variable definitions. Variables in common across domains have similar name extensions, and to make them standard, the beginning prefix of each variable should typically be a two-letter domain code.

PROC CDISC supports data validation for CDISC SDTM version 3.1.

---

## Validating a CDISC SDTM SAS Data Set

To validate a SAS data set that conforms to CDISC SDTM, submit the PROC CDISC statement and specify the MODEL=SDTM argument.

In addition, you must submit these statements:

- SDTM statement to specify the SDTM version number
- DOMAINDATA statement to specify the SAS data set, domain, and category

---

## CDISC Procedure Capabilities for Validating a CDISC SDTM SAS Data Set

### *Domain Content Validation*

PROC CDISC performs the following checks on domain content:

- Verifies that all required variables are in the SAS data set.
- Reports as an error any variables in the SAS data set that are not defined in the domain. Compared to CDISC SDS version 2.x, CDISC SDTM version 3.1 has more restrictions.
- Reports a warning for any expected domain variables that are not in the SAS data set. Compared to CDISC SDS version 2.x, CDISC SDTM version 3.1 has more required domain variables.
- Notes any permitted domain variables that are not in the SAS data set. PROC CDISC finds general omissions, but it is up to the site administrator to determine whether the omissions are appropriate.
- Verifies that all domain variables are of the expected data type and proper length. For example, the validation reports if the SAS data set has SAS date and time variables that require ISO 8601 expansion.
- Detects any domain variables that are assigned a controlled terminology specification by the domain and do not have a SAS format assigned to them. Only an assigned format is required.

PROC CDISC also performs the following checks on domain content on a per-observation basis:

- Verifies that all required variable fields do not contain missing values.
- Detects occurrences of expected variable fields that contain missing values.
- Detects the conformance of all ISO 8601 assigned values, including date, time, datetime, duration, and interval types.
- Notes correctness of YES or NO and YES, NO, or NULL responses.

With the exception of YES or NO and YES, NO, or NULL content, PROC CDISC does not validate the content of controlled terminology against a list of acceptable values.

### *Unsupported Operations*

PROC CDISC does not do the following operations:

- Automatically generate a V5 XPORT file from the SAS data set.
- Create an XML document. PROC CDISC performs SDTM 3.1 data content validation only.

## Part 2

---

# Usage

<i>Chapter 4</i>	
<b>CDISC Procedure Examples for CDISC ODM</b> .....	<b>17</b>
<i>Chapter 5</i>	
<b>CDISC Procedure Examples for CDISC SDTM</b> .....	<b>53</b>



## Chapter 4

# CDISC Procedure Examples for CDISC ODM

---

<b>Importing a CDISC ODM XML Document Using Default KeySet Processing . . .</b>	<b>18</b>
Overview . . . . .	18
Program . . . . .	18
Output . . . . .	19
<b>Importing a CDISC ODM XML Document Specifying KeySet Processing Options . . . . .</b>	<b>20</b>
Overview . . . . .	20
Program . . . . .	20
Output . . . . .	21
<b>Importing a CDISC ODM XML Document Using a Language Identifier . . . . .</b>	<b>22</b>
Overview . . . . .	22
Program . . . . .	25
Output . . . . .	26
<b>Importing a CDISC ODM XML Document with OrderNumber Attributes . . . . .</b>	<b>28</b>
Overview . . . . .	28
Program . . . . .	28
Output . . . . .	28
<b>Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax . . . . .</b>	<b>29</b>
Overview . . . . .	29
Program . . . . .	29
Output . . . . .	31
<b>Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets . . . . .</b>	<b>37</b>
Overview . . . . .	37
Program . . . . .	37
Output . . . . .	39
<b>Exporting a CDISC ODM XML Document with Required and Optional Statements . . . . .</b>	<b>42</b>
Overview . . . . .	42
Program . . . . .	42
<b>Describing a CDISC ODM SAS Data Set with the CONTENTS Statement . . . . .</b>	<b>45</b>
Overview . . . . .	45
Program with LONGNAMES=NO . . . . .	45
Output with LONGNAMES=NO . . . . .	47
Program with LONGNAMES=YES . . . . .	47
Output with LONGNAMES=YES . . . . .	49

<b>Listing a Directory with the DATASETS Statement</b> .....	<b>50</b>
Overview .....	50
Program with LONGNAMES=NO .....	50
Output with LONGNAMES=NO .....	51
Program with LONGNAMES=YES .....	51
Output with LONGNAMES=YES .....	52

---

## Importing a CDISC ODM XML Document Using Default KeySet Processing

### Overview

This example imports clinical trials data from a CDISC ODM XML document that is named AE.XML, and creates a SAS data set that is named RESULTS.AE.

The example illustrates the default behavior for KeySet members written to the output SAS data set. All KeySet members that are in the input XML document are converted to SAS variables and values in the output SAS data set.

To view the AE.XML document, see [“Sample CDISC ODM XML Document”](#) on page 89.

### Program

The following SAS program imports the XML document as a SAS data set:

1. The LIBNAME statement assigns the libref RESULTS to the physical location of the output SAS data set.
2. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.
3. The PROC CDISC statement specifies the following:
  - CDISC ODM as the model.
  - Fileref XMLINP, which references the physical location of the input XML document to be imported.
  - FORMATACTIVE=YES to convert CDISC ODM CodeList content in the XML document to SAS formats.
  - FORMATNOREPLACE=NO to replace existing SAS formats in the FORMAT catalog that have the same name as the converted formats.
4. ODMMINIMUMKEYSET=NO in the ODM statement specifies that all KeySet members are written to the output SAS data set. This is the default setting for ODMMINIMUMKEYSET=.
5. The CLINICALDATA statement identifies the output SAS data set, which is RESULTS.AE, and specifies the CDISC ODM ItemGroupDef attribute that indicates where the data content in the XML document begins, which is AE.
6. The CONTENTS procedure lists the contents of the output SAS data set. The VARNUM option lists the variables in the order in which they were created.



```

libname results 'C:\My Documents\'; 1

filename xmlinp 'C:\XML\ae.xml'; 2

proc cdisc model=odm 3
    read=xmlinp
    formatactive=yes
    formatnoreplace=no;
    odm odmversion="1.2" odmmminimumkeyset=no; 4
    clinicaldata out=results.AE sasdatasetname="AE"; 5
run;

proc contents data=results.AE varnum; 6
run;

filename xmlinp clear;

libname results clear;

```

## Output

The output from PROC CONTENTS displays the attributes of each interpreted variable, such as the variable's type and length. The attributes are obtained from the embedded metadata content.

Because ODMMINIMUMKEYSET=NO, all KeySet members are written to the output SAS data set. These are the first 10 variables listed in the output. (The first two characters in resulting SAS variable names are two underscores.)

The maximum OID length, which is 100, is allocated to each KeySet member.

**Display 4.1** PROC CONTENTS Output for RESULTS.AE

Variables in Creation Order					
#	Variable	Type	Len	Format	Informat
1	__StudyOID	Char	100		
2	__MetaDataVersionOID	Char	100		
3	__SubjectKey	Char	100		
4	__StudyEventOID	Char	100		
5	__StudyEventRepeatKey	Char	100		
6	__FormOID	Char	100		
7	__FormRepeatKey	Char	100		
8	__ItemGroupOID	Char	100		
9	__ItemGroupRepeatKey	Char	100		
10	__TransactionType	Char	100		

## Importing a CDISC ODM XML Document Specifying KeySet Processing Options

### Overview

This example imports clinical trials data from a CDISC ODM XML document that is named AE.XML, and creates a SAS data set that is named RESULTS.AEMIN.

The example illustrates the results of specifying the KeySet processing options ODMMINIMUMKEYSET=YES and ODMMAXIMUMOIDLENGTH= in the ODM statement. Only the unique SubjectKey member is written to the output SAS data set. The character length for the KeySet members is reduced from the default OID length.

To view the AE.XML document, see [“Sample CDISC ODM XML Document” on page 89](#).

### Program

The following SAS program imports the XML document as a SAS data set:

1. The LIBNAME statement assigns the libref RESULTS to the physical location of the output SAS data set.
2. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.
3. The PROC CDISC statement specifies the following:
  - CDISC ODM as the model.
  - Fileref XMLINP, which references the physical location of the input XML document to be imported.
  - FORMATACTIVE=YES to convert CDISC ODM CodeList content in the XML document to SAS formats.
  - FORMATNOREPLACE=NO to replace existing SAS formats in the FORMAT catalog that have the same name as the converted formats.
4. ODMMINIMUMKEYSET=YES in the ODM statement specifies that only the SubjectKey is written to the output SAS data set.  
ODMMAXIMUMOIDLENGTH=18 in the ODM statement allocates a storage space of 18 characters for the KeySet member character length, instead of the default maximum OID length.
5. The CLINICALDATA statement identifies the output SAS data set, which is RESULTS.AEMIN, and specifies the CDISC ODM ItemGroupDef attribute that indicates where the data content in the XML document begins, which is AE.
6. The CONTENTS procedure lists the contents of the output SAS data set. The VARNUM option lists the variables in the order in which they were created.

```
libname results 'C:\My Documents\'; 1
```

```
filename xmlinp 'C:\XML\ae.xml'; 2
```

```
proc cdisc model=odm 3
    read=xmlinp
    formatactive=yes
    formatnoreplace=no;

    odm odmversion="1.2"
    odmmimumkeyset=yes 4
    odmmaximumoidlength=18;

    clinicaldata out=results.AEMIN sasdatasetname="AE"; 5
run;

filename xmlinp clear;

proc contents data=results.AEMIN varnum; 6
run;

libname results clear;
```

## Output

The output from PROC CONTENTS displays the attributes of each interpreted variable, such as the variable's type and length. The attributes are obtained from the embedded metadata content.

Because ODMMINIMUMKEYSET=YES, only the SubjectKey is written to the output SAS data set, which is the first variable listed in the output.

Because ODMMAXIMUMOIDLENGTH=18, an OID length of 18 is allocated.

**Display 4.2** PROC CONTENTS Output for RESULTS.AEMIN

Variables in Creation Order					
#	Variable	Type	Len	Format	Informat
1	__SubjectKey	Char	18		
2	TAREA	Char	4	\$TAREAF.	
3	PNO	Char	15		
4	SCTRY	Char	4	\$SCTRYF.	
5	F_STATUS	Char	1	\$F_STATU.	
6	LINE_NO	Num	8		2.
7	AETERM	Char	100		
8	AESTMON	Num	8		2.
9	AESTDAY	Num	8		2.
10	AESTYR	Num	8		4.
11	AESTDT	Num	8	DATE.	
12	AEENMON	Num	8		2.
13	AEENDAY	Num	8		2.
14	AEENYR	Num	8		4.
15	AEENDT	Num	8	DATE.	
16	AESEV	Char	1	\$AESEV.	
17	AEREL	Char	1	\$AEREL.	
18	AEOUT	Char	1	\$AEOUT.	
19	AEACTTRT	Char	1	\$AEACTTR.	
20	AECONTRT	Char	1	\$AECONTR.	

---

## Importing a CDISC ODM XML Document Using a Language Identifier

### Overview

This example imports clinical trials data from a CDISC ODM XML document by specifying a language identifier with the LANGUAGE= option in the PROC CDISC statement. By specifying the LANGUAGE= option, PROC CDISC locates the matching language identifier in the ODM TranslatedText element. It creates a SAS format by using the TranslatedText items with a matching language tag attribute (xml:lang). The created SAS format is then applied to the data that is imported from the XML document.

This example imports the following XML document:

```
<?xml version="1.0" encoding="windows-1252" ?>
<!--
    Clinical Data Interchange Standards Consortium (CDISC)
    Operational Data Model (ODM) for clinical data interchange

    You can learn more about CDISC standards efforts at
    http://www.cdisc.org/standards/index.html
-->

<ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 ODM1-2-0.xsd"

      ODMVersion="1.2"
      FileOID="000-00-0000"
      FileType="Snapshot"
      Description="testing codelist stuff"

      AsOfDateTime="2006-11-03T09:47:53"
      CreationDateTime="2006-11-03T09:47:53"
      SourceSystem="SAS"
      SourceSystemVersion="GENERIC"
      >

<Study OID="AStudyOID">

  <!--
    GlobalVariables is a REQUIRED section in ODM markup
  -->
  <GlobalVariables>
    <StudyName>CODELIST</StudyName>
    <StudyDescription>Checking Codelists</StudyDescription>
    <ProtocolName>Protocol</ProtocolName>
  </GlobalVariables>

  <BasicDefinitions />

  <!--
    Internal ODM markup required metadata
  -->
  <MetaDataVersion OID="MDV_CODELIST" Name="MDV Codelist">
    <Protocol>
      <StudyEventRef StudyEventOID="StudyEventOID" OrderNumber="1"
        Mandatory="Yes" />
    </Protocol>

    <StudyEventDef OID="StudyEventOID" Name="Study Event Definition"
      Repeating="Yes" Type="Common">
      <FormRef FormOID="X" OrderNumber="1" Mandatory="No" />
    </StudyEventDef>

    <FormDef OID="X" Name="Form Definition" Repeating="Yes">
      <ItemGroupRef ItemGroupOID="X" Mandatory="No" />
    </FormDef>
  </MetaDataVersion>
</Study>

```

```

</FormDef>

<!--
    Columns defined in the table
-->
<ItemGroupDef OID="X" Repeating="Yes"
    SASDatasetName="X"
    Name="ODM Examples"
    Comment="Examples of ODM Datatypes">
    <ItemRef ItemOID="ID.x" OrderNumber="1" Mandatory="No" />
</ItemGroupDef>

<!--
    Column attributes as defined in the table
-->
<ItemDef OID="ID.x" SASFieldName="x" Name="x" DataType="float" Length="12"
    SignificantDigits="2" Comment="x">
    <CodeListRef CodeListOID="CL.NUMBERS" />
</ItemDef>

<!--
    Translation to ODM markup for any PROC FORMAT style
    user defined or SAS internal formatting specifications
    applied to columns in the table
-->
<CodeList OID="CL.NUMBERS" SASFormatName="NUMBERS" Name="NUMBERS"
    DataType="float">
    <CodeListItem CodedValue="1">
        <Decode>
            <TranslatedText xml:lang="de">einz</TranslatedText>
            <TranslatedText xml:lang="en">one</TranslatedText>
            <TranslatedText xml:lang="es">uno</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="2">
        <Decode>
            <TranslatedText xml:lang="de">zwei</TranslatedText>
            <TranslatedText xml:lang="en">two</TranslatedText>
            <TranslatedText xml:lang="es">dos</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="3">
        <Decode>
            <TranslatedText xml:lang="de">drei</TranslatedText>
            <TranslatedText xml:lang="en">three</TranslatedText>
            <TranslatedText xml:lang="es">tres</TranslatedText>
        </Decode>
    </CodeListItem>
</CodeList>
</MetaDataVersion>
</Study>

```

```

<!--
    Administrative metadata
-->
<AdminData />

<!--
    Actual data content begins here
    This section represents each data record in the table
-->
<ClinicalData StudyOID="AStudyOID" MetaDataVersionOID="MDV_CODELIST">
  <SubjectData SubjectKey="001">
    <StudyEventData StudyEventOID="StudyEventOID" StudyEventRepeatKey="1">
      <FormData FormOID="X" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="X" ItemGroupRepeatKey="1">
          <ItemData ItemOID="ID.x" Value="3" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>

```

## Program

The following SAS program imports the XML document as a SAS data set:

1. The LIBNAME statement assigns the libref RESULTS to the physical location of the output SAS data set.
2. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.
3. The PROC CDISC statement specifies the following:
  - CDISC ODM as the model.
  - Fileref XMLINP, which references the physical location of the input XML document to be imported.
  - FORMATACTIVE=YES to convert CDISC ODM CodeList content in the XML document to SAS formats.
  - FORMATNOREPLACE=NO to replace existing SAS formats in the FORMAT catalog that have the same name as the converted formats.
  - LANGUAGE="DE" to specify a language identifier with a two-letter language code. PROC CDISC locates the DE language identifier in the ODM TranslatedText element and creates a SAS format by using the TranslatedText items with a matching language tag attribute. The created SAS format is then applied to the data that is imported from the XML document.
4. ODDMINIMUMKEYSET=NO in the ODM statement specifies that all KeySet members are written to the output SAS data set. This is the default setting for ODDMINIMUMKEYSET=.
5. The CLINICALDATA statement identifies the output SAS data set, which is RESULTS.NUMBERS, and specifies the CDISC ODM ItemGroupDef attribute that indicates where the data content in the XML document begins, which is X.

6. The CONTENTS procedure lists the contents of the output SAS data set.
7. The PRINT procedure prints the rows in the output SAS data set. The VAR statement selects just the X variable.

```

libname results 'C:\My Documents\'; 1

filename xmlinp 'C:\XML\numbers.xml'; 2

proc cdisc model=odm 3
    read=xmlinp
    formatactive=yes
    formatnoreplace=no
    language="de";
    odm odmversion="1.2" odminimumkeyset=no; 4
    clinicaldata out=results.numbers sasdatasetname="X"; 5
run;

filename xmlinp clear;

proc contents data=results.numbers; 6
run;

proc print data=results.numbers; 7
    var x;
run;

libname results clear;

```

## Output

The output from PROC CONTENTS displays the attributes of each interpreted variable, which includes the SAS variable X and all KeySet members.



**Display 4.3** PROC CONTENTS Output for RESULTS.NUMBERS

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
11	X	Num	8	NUMBERS.	12.2	x
6	__FormOID	Char	100			
7	__FormRepeatKey	Char	100			
8	__ItemGroupOID	Char	100			
9	__ItemGroupRepeatKey	Char	100			
2	__MetaDataVersionOID	Char	100			
4	__StudyEventOID	Char	100			
5	__StudyEventRepeatKey	Char	100			
1	__StudyOID	Char	100			
3	__SubjectKey	Char	100			
10	__TransactionType	Char	100			

The output from PROC PRINT lists the value for the imported SAS variable X. The procedure applies the SAS format NUMBERS, which is created by using the TranslatedText item with the matching language tag attribute DE. It applies NUMBERS to the data that is imported from the XML document, which is 3. The result is the German word drei.

**Display 4.4** PROC PRINT Output for Variable X

The screenshot shows a window titled "The SAS System" with a table of data. The table has two columns: "Obs" and "X". The first row shows "1" under "Obs" and "drei" under "X".

Obs	X
1	drei

---

## Importing a CDISC ODM XML Document with OrderNumber Attributes

### Overview

This example imports a CDISC ODM XML document that contains OrderNumber attributes that are not in sequence. PROC CDISC validates the attributes and displays warnings in the SAS log.

### Program

The following SAS program imports the XML document as a SAS data set:

1. The LIBNAME statement assigns the libref RESULTS to the physical location of the output SAS data set.
2. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.
3. The PROC CDISC statement specifies the following:
  - CDISC ODM as the model.
  - Fileref XMLINP, which references the physical location of the input XML document to be imported.
4. ORDERNUMBER=YES in the ODM statement specifies to validate whether OrderNumber attributes conform to the model specifications. This is the default setting for ORDERNUMBER=.
5. The CLINICALDATA statement identifies the output SAS data set, which is RESULTS.AE, and specifies the CDISC ODM ItemGroupDef attribute that indicates where the data content in the XML document begins, which is AE.

```
libname results 'C:\My Documents\'; 1

filename xmlinp 'C:\XML\ae.xml'; 2

proc cdisc model=odm read=xmlinp; 3
  odm odmversion="1.2" ordernumber=yes; 4
  clinicaldata out=results.AE sasdatasetname="AE"; 5
run;

filename xmlinp clear;

libname results clear;
```

### Output

The following SAS log displays a warning that the ItemRef element contains an OrderNumber attribute that is out of sequence.

```
238 libname results 'C:\My Documents\';
NOTE: Libref RESULTS was successfully assigned as follows:
      Engine:          V9
      Physical Name:  C:\My Documents\
239
240 filename xmlinp 'C:\XML\ae.xml';
241
242 proc cdisc model=odm read=xmlinp;
243     odm odmversion="1.2" ordernumber=yes;
244     clinicaldata out=results.AE sasdatasetname="AE";
245 run;

WARNING: ItemRef OID="ID.AESEV" OrderNumber="21" outside range. Ignoring
OrderNumber.
NOTE: PROCEDURE CDISC used (Total process time):
      real time          4.76 seconds
      cpu time           0.01 seconds
```

---

## Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax

### Overview

This example, which includes only the required PROC CDISC statements for exporting, specifies the metadata attributes in the statements.

As an alternative, you can store metadata attributes for the required ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION statements in separate SAS data sets that you reference with the DATA= argument. For an example, see [“Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets”](#) on page 37.

### Program

The following SAS program exports a SAS data set as a CDISC ODM XML document:

1. The LIBNAME statement assigns the libref RESULTS to the physical location of the input SAS data set.
2. The FILENAME statement assigns the fileref XMLOUT to the physical location of the output XML document (complete pathname, filename, and file extension) to be exported.
3. The SORT procedure specifies to sort the input SAS data set by the `__SUBJECTKEY` variable.
4. The PROC CDISC statement specifies the following:
  - CDISC ODM as the model.
  - Fileref XMLOUT, which references the physical location of the output XML document to be exported.
  - FORMATACTIVE=YES to convert CDISC ODM CodeList content in the XML document to SAS formats.

- FORMATNOREPLACE=NO to replace existing SAS formats in the FORMAT catalog that have the same name as the converted formats.
5. The ODM statement specifies the required metadata attributes for exporting, such as the CDISC ODM version and file type.
  6. The STUDY statement specifies the study identifier.
  7. Options in the GLOBALVARIABLES statement specify general summary information about the study.
  8. The METADATAVERSION statement specifies the required metadata version and version name.
  9. The CLINICALDATA statement identifies the input SAS data set, which is RESULTS.AE. The input SAS data set contains the data content and KeySet members that are written to the XML document. In addition, the CLINICALDATA statement specifies optional metadata attributes.

```

libname results 'C:\My Documents\'; 1

filename xmlout 'C:\XML\AEfull.XML'; 2

proc sort data=results.AE; 3
  by __subjectkey;
run;

proc cdisc model=odm 4
  write=xmlout
  formatactive=yes
  formatnoreplace=no;

  odm odmversion="1.2" 5
  fileoid="000-00-0000"
  filetype=SNAPSHOT
  description="Adverse events from the CTChicago file";

  study studyoid="STUDY.StudyOID"; 6

  globalvariables studyname="CDISC Connect-A-Thon Test Study III" 7
  studydescription="This file contains test data for the CDISC
  Connect-A-Thon event scheduled for the DIA
  38th annual meeting in Chicago."
  protocolname="CDISC-Protocol-00-000";

  metadataversion metadataversionoid="v1.1.0" 8
  name="Version 1.1.0";

  clinicaldata data=results.AE 9
  domain="AE"
  name="Adverse Events"
  comment="All adverse events in this trial" ;
run;

filename xmlout clear;

libname results clear;

```

## **Output**

The following is an annotated excerpt of the output CDISC ODM XML document:

1. All of the metadata is contained by a single `MetaDataVersion` element.
2. A `Protocol` element that contains a `StudyEventRef` for each `StudyEventDef` is automatically generated.
3. A `StudyEventDef` element that contains a `FormRef` for each `FormDef` is automatically generated.
4. A `FormDef` element that contains an `ItemGroupRef` for each `ItemGroupDef` is automatically generated.
5. The SAS data set is represented as an `ItemGroupDef`.
6. Each variable in the SAS data set is represented as an `ItemRef`.
7. If a variable contains a reference to a user-defined SAS format, a `CodeListRef` is generated in the `ItemDef` for the variable.
8. A `CodeList` element is generated for each unique user-defined SAS format that is referenced.
9. A `ClinicalData` element is created for each `ItemGroupDef` that is referenced.

**Output 4.1** CDISC ODM XML Document Exported by PROC CDISC

```

<?xml version="1.0" encoding="windows-1252" ?>
<!--
    Clinical Data Interchange Standards Consortium (CDISC)
    Operational Data Model (ODM) for clinical data interchange

    You can learn more about CDISC standards efforts at
    http://www.cdisc.org/standards/index.html
-->

<ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
    xmlns:ds="http://www.w3.org/2000/09/xmlsig#"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 ODM1-2-0.xsd"

    ODMVersion="1.2"
    FileOID="000-00-0000"
    FileType="Snapshot"
    Description="Adverse events from the CTChicago file"

    AsOfDateTime="2010-09-30T14:07:31"
    CreationDateTime="2010-09-30T14:07:31"
    SourceSystem="SAS 9.3 PROC CDISC"
    SourceSystemVersion="9.03.01B0D09292010 3.00.65">

<Study OID="STUDY.StudyOID">

<!--
    GlobalVariables is a REQUIRED section in ODM markup
-->
<GlobalVariables>
<StudyName>CDISC Connect-A-Thon Test Study III</StudyName>
<StudyDescription>This file contains test data for the CDISC Connect-A-Thon event
scheduled for the DIA 38th annual meeting in Chicago.</StudyDescription>
<ProtocolName>CDISC-Protocol-00-000</ProtocolName>
</GlobalVariables>

<BasicDefinitions />

<!--
    Internal ODM markup required metadata
-->
<MetaDataVersion OID="v1.1.0" Name="Version 1.1.0"> 1
</Protocol> 2
<StudyEventRef StudyEventOID="SE.VISIT1" OrderNumber="1" Mandatory="Yes" />
</Protocol>

<StudyEventDef OID="SE.VISIT1" Name="Study Event Definition" Repeating="Yes" Type="Common"> 3
<FormRef FormOID="FORM.AE" OrderNumber="1" Mandatory="No" />
</StudyEventDef>

<FormDef OID="FORM.AE" Name="Form Definition" Repeating="Yes"> 4
<ItemGroupRef ItemGroupOID="IG.AE" Mandatory="No" />
</FormDef>

<!--
    Columns defined in the table
-->
<ItemGroupDef OID="IG.AE" Repeating="Yes"> 5
    SASDatasetName="AE"
    Name="Adverse Events"
    Domain="AE"
    Comment="All adverse events in this trial">

```

```

<ItemRef ItemOID="ID.TAREA" OrderNumber="1" Mandatory="No" /> 6
<ItemRef ItemOID="ID.PNO" OrderNumber="2" Mandatory="No" />
<ItemRef ItemOID="ID.SCTRY" OrderNumber="3" Mandatory="No" />
<ItemRef ItemOID="ID.F_STATUS" OrderNumber="4" Mandatory="No" />
<ItemRef ItemOID="ID.LINE_NO" OrderNumber="5" Mandatory="No" />
<ItemRef ItemOID="ID.AETERM" OrderNumber="6" Mandatory="No" />
<ItemRef ItemOID="ID.AESTMON" OrderNumber="7" Mandatory="No" />
<ItemRef ItemOID="ID.AESTDAY" OrderNumber="8" Mandatory="No" />
<ItemRef ItemOID="ID.AESTYR" OrderNumber="9" Mandatory="No" />
<ItemRef ItemOID="ID.AESTDT" OrderNumber="10" Mandatory="No" />
<ItemRef ItemOID="ID.AEENMON" OrderNumber="11" Mandatory="No" />
<ItemRef ItemOID="ID.AEENDAY" OrderNumber="12" Mandatory="No" />
<ItemRef ItemOID="ID.AEENYR" OrderNumber="13" Mandatory="No" />
<ItemRef ItemOID="ID.AEENDT" OrderNumber="14" Mandatory="No" />
<ItemRef ItemOID="ID.AESEV" OrderNumber="15" Mandatory="No" />
<ItemRef ItemOID="ID.AEREL" OrderNumber="16" Mandatory="No" />
<ItemRef ItemOID="ID.AEOUT" OrderNumber="17" Mandatory="No" />
<ItemRef ItemOID="ID.AEACTTRT" OrderNumber="18" Mandatory="No" />
<ItemRef ItemOID="ID.AECONTRT" OrderNumber="19" Mandatory="No" />
</ItemGroupDef>

<!--
  Column attributes as defined in the table
-->
<ItemDef OID="ID.TAREA" Name="TAREA" SASFieldName="TAREA" DataType="text" Length="4">
<CodeListRef CodeListOID="CL.$TAREAF" /> 7
</ItemDef>
<ItemDef OID="ID.PNO" Name="PNO" SASFieldName="PNO" DataType="text" Length="15" />
<ItemDef OID="ID.SCTRY" Name="SCTRY" SASFieldName="SCTRY" DataType="text" Length="4">
<CodeListRef CodeListOID="CL.$SCTRYF" />
</ItemDef>
<ItemDef OID="ID.F_STATUS" Name="F_STATUS" SASFieldName="F_STATUS" DataType="text" Length="1">
<CodeListRef CodeListOID="CL.$F_STATU" />
</ItemDef>
<ItemDef OID="ID.LINE_NO" Name="LINE_NO" SASFieldName="LINE_NO" DataType="integer" Length="2" />
<ItemDef OID="ID.AETERM" Name="AETERM" SASFieldName="AETERM" DataType="text" Length="100" />
<ItemDef OID="ID.AESTMON" Name="AESTMON" SASFieldName="AESTMON" DataType="integer" Length="2" />
<ItemDef OID="ID.AESTDAY" Name="AESTDAY" SASFieldName="AESTDAY" DataType="integer" Length="2" />
<ItemDef OID="ID.AESTYR" Name="AESTYR" SASFieldName="AESTYR" DataType="integer" Length="4" />
<ItemDef OID="ID.AESTDT" Name="AESTDT" SASFieldName="AESTDT" DataType="date" />
<ItemDef OID="ID.AEENMON" Name="AEENMON" SASFieldName="AEENMON" DataType="integer" Length="2" />
<ItemDef OID="ID.AEENDAY" Name="AEENDAY" SASFieldName="AEENDAY" DataType="integer" Length="2" />
<ItemDef OID="ID.AEENYR" Name="AEENYR" SASFieldName="AEENYR" DataType="integer" Length="4" />
<ItemDef OID="ID.AEENDT" Name="AEENDT" SASFieldName="AEENDT" DataType="date" />
<ItemDef OID="ID.AESEV" Name="AESEV" SASFieldName="AESEV" DataType="text" Length="1">
<CodeListRef CodeListOID="CL.$AESEV" />
</ItemDef>
<ItemDef OID="ID.AEREL" Name="AEREL" SASFieldName="AEREL" DataType="text" Length="1">
<CodeListRef CodeListOID="CL.$AEREL" />
</ItemDef>
<ItemDef OID="ID.AEOUT" Name="AEOUT" SASFieldName="AEOUT" DataType="text" Length="1">
<CodeListRef CodeListOID="CL.$AEOUT" />
</ItemDef>
<ItemDef OID="ID.AEACTTRT" Name="AEACTTRT" SASFieldName="AEACTTRT" DataType="text" Length="1">
<CodeListRef CodeListOID="CL.$AEACTTR" />
</ItemDef>
<ItemDef OID="ID.AECONTRT" Name="AECONTRT" SASFieldName="AECONTRT" DataType="text" Length="1">
<CodeListRef CodeListOID="CL.$AECONTR" />
</ItemDef>

```

```

<!--
  Translation to ODM markup for any PROC FORMAT style
  user defined or SAS internal formatting specifications
  applied to columns in the table
-->
<CodeList OID="CL.$TAREAF" SASFormatName="$TAREAF" Name="$TAREAF" DataType="text">
  <CodeListItem CodedValue='ONC'>
    <Decode>
      <TranslatedText>Oncology</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

<CodeList OID="CL.$SCTRYF" SASFormatName="$SCTRYF" Name="$SCTRYF" DataType="text">
  <CodeListItem CodedValue='USA'>
    <Decode>
      <TranslatedText>United States</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

<CodeList OID="CL.$F_STATU" SASFormatName="$F_STATU" Name="$F_STATU" DataType="text">
  <CodeListItem CodedValue='S'>
    <Decode>
      <TranslatedText>Source verified, not queried</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='V'>
    <Decode>
      <TranslatedText>Source verified, queried</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

<CodeList OID="CL.$AESEV" SASFormatName="$AESEV" Name="$AESEV" DataType="text">
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText>Mild</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText>Moderate</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText>Severe</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='4'>
    <Decode>
      <TranslatedText>Life Threatening</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

```



```
<CodeList OID="CL.$AEREL" SASFormatName="$AEREL" Name="$AEREL" DataType="text">
<CodeListItem CodedValue='0'>
<Decode>
<TranslatedText>None</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='1'>
<Decode>
<TranslatedText>Unlikely</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='2'>
<Decode>
<TranslatedText>Possible</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='3'>
<Decode>
<TranslatedText>Probable</TranslatedText>
</Decode>
</CodeListItem>
</CodeList>

<CodeList OID="CL.$AEOUT" SASFormatName="$AEOUT" Name="$AEOUT" DataType="text">
<CodeListItem CodedValue='1'>
<Decode>
<TranslatedText>Resolved, no residual effects</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='2'>
<Decode>
<TranslatedText>Continuing</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='3'>
<Decode>
<TranslatedText>Resolved, residual effects</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='4'>
<Decode>
<TranslatedText>Death</TranslatedText>
</Decode>
</CodeListItem>
</CodeList>

<CodeList OID="CL.$AEACTTR" SASFormatName="$AEACTTR" Name="$AEACTTR" DataType="text">
<CodeListItem CodedValue='0'>
<Decode>
<TranslatedText>None</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='1'>
<Decode>
<TranslatedText>Discontinued permanently</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='2'>
<Decode>
<TranslatedText>Reduced</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='3'>
<Decode>
<TranslatedText>Interrupted</TranslatedText>
</Decode>
</CodeListItem>
</CodeList>
```

```

<CodeList OID="CL.$AECONTR" SASFormatName="$AECONTR" Name="$AECONTR" DataType="text">
<CodeListItem CodedValue='0'>
<Decode>
<TranslatedText>None</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='1'>
<Decode>
<TranslatedText>Medication required</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='2'>
<Decode>
<TranslatedText>Hospitalization required or prolonged</TranslatedText>
</Decode>
</CodeListItem>
<CodeListItem CodedValue='3'>
<Decode>
<TranslatedText>Other</TranslatedText>
</Decode>
</CodeListItem>
</CodeList>
</MetaDataVersion>
</Study>

<!--
    Administrative metadata
-->
<AdminData />

<!--
    Actual data content begins here
    This section represents each data record in the table
-->
<ClinicalData StudyOID="STUDY.StudyOID" MetaDataVersionOID="v1.1.0">
<SubjectData SubjectKey="001">
<StudyEventData StudyEventOID="SE.VISIT1" StudyEventRepeatKey="1">
<FormData FormOID="FORM.AE" FormRepeatKey="1">
<ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="1">
<ItemData ItemOID="ID.TAREA" Value="ONC" />
<ItemData ItemOID="ID.PNO" Value="143-02" />
<ItemData ItemOID="ID.SCTRY" Value="USA" />
<ItemData ItemOID="ID.F_STATUS" Value="V" />
<ItemData ItemOID="ID.LINE_NO" Value="1" />
<ItemData ItemOID="ID.AETERM" Value="HEADACHE" />
<ItemData ItemOID="ID.AESTMON" Value="6" />
<ItemData ItemOID="ID.AESTDAY" Value="10" />
<ItemData ItemOID="ID.AESTYR" Value="1999" />
<ItemData ItemOID="ID.AESTDT" Value="1999-06-10" />
<ItemData ItemOID="ID.AEENMON" Value="6" />
<ItemData ItemOID="ID.AEENDAY" Value="14" />
<ItemData ItemOID="ID.AEENYR" Value="1999" />
<ItemData ItemOID="ID.AEENDT" Value="1999-06-14" />
<ItemData ItemOID="ID.AESEV" Value="1" />
<ItemData ItemOID="ID.AEREL" Value="0" />
<ItemData ItemOID="ID.AEOUT" Value="1" />
<ItemData ItemOID="ID.AEACTTRT" Value="0" />
<ItemData ItemOID="ID.AECONTRT" Value="1" />
</ItemGroupData>

```

```

<ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="2">
<ItemData ItemOID="ID.TAREA" Value="ONC" />
<ItemData ItemOID="ID.PNO" Value="143-02" />
<ItemData ItemOID="ID.SCTRY" Value="USA" />
<ItemData ItemOID="ID.F_STATUS" Value="V" />
<ItemData ItemOID="ID.LINE_NO" Value="2" />
<ItemData ItemOID="ID.AETERM" Value="CONGESTION" />
<ItemData ItemOID="ID.AESTMON" Value="6" />
<ItemData ItemOID="ID.AESTDAY" Value="11" />
<ItemData ItemOID="ID.AESTYR" Value="1999" />
<ItemData ItemOID="ID.AESTDT" Value="1999-06-11" />
<ItemData ItemOID="ID.AEENMON" IsNull="Yes" />
<ItemData ItemOID="ID.AEENDAY" IsNull="Yes" />
<ItemData ItemOID="ID.AEENYR" IsNull="Yes" />
<ItemData ItemOID="ID.AEENDT" IsNull="Yes" />
<ItemData ItemOID="ID.AESEV" Value="1" />
<ItemData ItemOID="ID.AEREL" Value="0" />
<ItemData ItemOID="ID.AEOUT" Value="2" />
<ItemData ItemOID="ID.AEACTTRT" Value="0" />
<ItemData ItemOID="ID.AECONTRT" Value="1" />
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>

```

---

## Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets

### Overview

This example, which includes only the required PROC CDISC statements for exporting, illustrates how to specify metadata attributes that are stored in SAS data sets and referenced in the DATA= argument. For information about the advantages of using this method, see [“Specifying CDISC ODM Metadata Attributes” on page 9](#).

When exporting, you can specify metadata attributes for the required ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION statements either directly in the CDISC procedure statement or stored in a SAS data set. However, the optional metadata attributes for the required CLINICALDATA statement must be specified as part of the CLINICALDATA statement syntax.

### Program

First, create SAS data sets that contain metadata attributes. The LIBNAME statement assigns the libref CURRENT to the physical location of the SAS data sets that will store the metadata attributes.

```

libname current 'C:\mydata\';

data current.odm;
  odmversion="1.2";
  fileoid="ODM.FileOID";
  filetype="Snapshot";

```

```

        description="Optional descriptive information";
run;

data current.study;
    studyoid="OID.StudyOID";
run;

data current.globals;
    studyname="CDISC Test Study III";
    studydescription="This file contains test data for CDISC testing";
    protocolname="CDISC-Protocol-00-0000";
run;

data current.metadata;
    metadataversionoid="v1.1.0";
    name="Version 1.1.0";
run;

```

Then, reference the SAS data sets in a PROC CDISC program:

1. The FILENAME statement assigns the fileref XMLOUT to the physical location of the output XML document.
2. The required PROC CDISC statement specifies CDISC ODM as the model and the fileref XMLOUT, which references the physical location of the output XML document.
3. The required ODM statement includes the DATA= argument to reference the SAS data set CURRENT.ODM, which stores the CDISC ODM version and file type.
4. The required STUDY statement includes the DATA= argument to reference the SAS data set CURRENT.STUDY, which stores the study identifier.
5. The required GLOBALVARIABLES statement includes the DATA= argument to reference the SAS data set CURRENT.GLOBALS, which stores general summary information about the study.
6. The required METADATAVERSION statement includes the DATA= argument to reference the SAS data set CURRENT.METADATA, which stores the metadata version and version name.
7. The required CLINICALDATA statement identifies the input SAS data set, which is CURRENT.AE. The input SAS data set contains the data content and KeySet members that are written to the XML document. In addition, the CLINICALDATA statement specifies optional metadata attributes.

```

filename xmlout 'C:\XML\aeds.xml'; 1

proc cdisc model=odm write=xmlout; 2
    odm data=current.odm; 3
    study data=current.study; 4
    globalvariables data=current.globals; 5
    metadataversion data=current.metadata; 6
    clinicaldata data = current.ae 7
        domain="AE"
        name="Adverse Events"
        comment="All adverse events in this trial";
run;

filename xmlout clear;

```

## Output

The following output shows the exported XML document:

### Output 4.2 CDISC ODM XML Document Exported by PROC CDISC

```
<?xml version="1.0" encoding="windows-1252" ?>
<!--
  Clinical Data Interchange Standards Consortium (CDISC)
  Operational Data Model (ODM) for clinical data interchange

  You can learn more about CDISC standards efforts at
  http://www.cdisc.org/standards/index.html
-->

<ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 ODM1-2-0.xsd"

  ODMVersion="1.2"
  FileOID="ODM.FileOID"
  FileType="Snapshot"
  Description="Optional descriptive information"

  AsOfDateTime="2009-03-09T12:45:33"
  CreationDateTime="2009-03-09T12:45:33"
  SourceSystem="SAS 9.2 PROC CDISC"
  SourceSystemVersion="9.02.02M0D01222009 3.00.65">

  <Study OID="STUDY.StudyOID">

    <!--
      GlobalVariables is a REQUIRED section in ODM markup
    -->
    <GlobalVariables>
      <StudyName>CDISC Test Study III</StudyName>
      <StudyDescription>This file contains test data for CDISC
testing</StudyDescription>
      <ProtocolName>CDISC-Protocol-00-0000</ProtocolName>
    </GlobalVariables>

    <BasicDefinitions />

    <!--
      Internal ODM markup required metadata
    -->
    <MetaDataVersion OID="v1.1.0" Name="Version 1.1.0">
      <Protocol>
        <StudyEventRef StudyEventOID="SE.VISIT1" OrderNumber="1"
Mandatory="Yes" />
      </Protocol>

      <StudyEventDef OID="SE.VISIT1" Name="Study Event Definition"
Repeating="Yes" Type="Common">
        <FormRef FormOID="FORM.AE" OrderNumber="1" Mandatory="No" />
      </StudyEventDef>

      <FormDef OID="FORM.AE" Name="Form Definition" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="IG.AE" Mandatory="No" />
      </FormDef>
```

```

<!--
    Columns defined in the table
-->
<ItemGroupDef OID="IG.AE" Repeating="Yes"
    SASDatasetName="AE"
    Name="Adverse Events"
    Domain="AE"
    Comment="All adverse events in this trial">

    <ItemRef ItemOID="ID.TAREA" OrderNumber="1" Mandatory="No" />
    <ItemRef ItemOID="ID.PNO" OrderNumber="2" Mandatory="No" />
    <ItemRef ItemOID="ID.SCTRY" OrderNumber="3" Mandatory="No" />
    <ItemRef ItemOID="ID.F_STATUS" OrderNumber="4" Mandatory="No" />
    <ItemRef ItemOID="ID.LINE_NO" OrderNumber="5" Mandatory="No" />
    <ItemRef ItemOID="ID.AETERM" OrderNumber="6" Mandatory="No" />
    <ItemRef ItemOID="ID.AESTMON" OrderNumber="7" Mandatory="No" />
    <ItemRef ItemOID="ID.AESTDAY" OrderNumber="8" Mandatory="No" />
    <ItemRef ItemOID="ID.AESTYR" OrderNumber="9" Mandatory="No" />
    <ItemRef ItemOID="ID.AESTDT" OrderNumber="10" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENMON" OrderNumber="11" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENDAY" OrderNumber="12" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENYR" OrderNumber="13" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENDT" OrderNumber="14" Mandatory="No" />
    <ItemRef ItemOID="ID.AESEV" OrderNumber="15" Mandatory="No" />
    <ItemRef ItemOID="ID.AEREL" OrderNumber="16" Mandatory="No" />
    <ItemRef ItemOID="ID.AEOUT" OrderNumber="17" Mandatory="No" />
    <ItemRef ItemOID="ID.AEACTTRT" OrderNumber="18" Mandatory="No" />
    <ItemRef ItemOID="ID.AECONTRT" OrderNumber="19" Mandatory="No" />
</ItemGroupDef>

<!--
    Column attributes as defined in the table
-->
<ItemDef OID="ID.TAREA" Name="TAREA" SASFieldName="TAREA"
DataType="text" Length="4" />
<ItemDef OID="ID.PNO" Name="PNO" SASFieldName="PNO" DataType="text"
Length="15" />
<ItemDef OID="ID.SCTRY" Name="SCTRY" SASFieldName="SCTRY"
DataType="text" Length="4" />
<ItemDef OID="ID.F_STATUS" Name="F_STATUS" SASFieldName="F_STATUS"
DataType="text" Length="1" />
<ItemDef OID="ID.LINE_NO" Name="LINE_NO" SASFieldName="LINE_NO"
DataType="integer" Length="2" />
<ItemDef OID="ID.AETERM" Name="AETERM" SASFieldName="AETERM"
DataType="text" Length="100" />
<ItemDef OID="ID.AESTMON" Name="AESTMON" SASFieldName="AESTMON"
DataType="integer" Length="2" />
<ItemDef OID="ID.AESTDAY" Name="AESTDAY" SASFieldName="AESTDAY"
DataType="integer" Length="2" />
<ItemDef OID="ID.AESTYR" Name="AESTYR" SASFieldName="AESTYR"
DataType="integer" Length="4" />
<ItemDef OID="ID.AESTDT" Name="AESTDT" SASFieldName="AESTDT"
DataType="date" />
<ItemDef OID="ID.AEENMON" Name="AEENMON" SASFieldName="AEENMON"
DataType="integer" Length="2" />
<ItemDef OID="ID.AEENDAY" Name="AEENDAY" SASFieldName="AEENDAY"
DataType="integer" Length="2" />
<ItemDef OID="ID.AEENYR" Name="AEENYR" SASFieldName="AEENYR"
DataType="integer" Length="4" />
<ItemDef OID="ID.AEENDT" Name="AEENDT" SASFieldName="AEENDT"
DataType="date" />
<ItemDef OID="ID.AESEV" Name="AESEV" SASFieldName="AESEV"
DataType="text" Length="1" />
<ItemDef OID="ID.AEREL" Name="AEREL" SASFieldName="AEREL"
DataType="text" Length="1" />

```

```

        <ItemDef OID="ID.AEOUT" Name="AEOUT" SASFieldName="AEOUT"
DataType="text" Length="1" />
        <ItemDef OID="ID.AEACTTRT" Name="AEACTTRT" SASFieldName="AEACTTRT"
DataType="text" Length="1" />
        <ItemDef OID="ID.AECONTRT" Name="AECONTRT" SASFieldName="AECONTRT"
DataType="text" Length="1" />
    </MetaDataVersion>
</Study>
<!--
    Administrative metadata
-->
<AdminData />

<!--
    Actual data content begins here
    This section represents each data record in the table
-->
<ClinicalData StudyOID="STUDY.StudyOID" MetaDataVersionOID="v1.1.0">
    <SubjectData SubjectKey="001">
        <StudyEventData StudyEventOID="SE.VISIT1" StudyEventRepeatKey="1">
            <FormData FormOID="FORM.AE" FormRepeatKey="1">
                <ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="1">
                    <ItemData ItemOID="ID.TAREA" Value="ONC" />
                    <ItemData ItemOID="ID.PNO" Value="143-02" />
                    <ItemData ItemOID="ID.SCTRY" Value="USA" />
                    <ItemData ItemOID="ID.F_STATUS" Value="V" />
                    <ItemData ItemOID="ID.LINE_NO" Value="1" />
                    <ItemData ItemOID="ID.AETERM" Value="HEADACHE" />
                    <ItemData ItemOID="ID.AESTMON" Value="6" />
                    <ItemData ItemOID="ID.AESTDAY" Value="10" />
                    <ItemData ItemOID="ID.AESTYR" Value="1999" />
                    <ItemData ItemOID="ID.AESTDT" Value="1999-06-10" />
                    <ItemData ItemOID="ID.AEENMON" Value="6" />
                    <ItemData ItemOID="ID.AEENDAY" Value="14" />
                    <ItemData ItemOID="ID.AEENYR" Value="1999" />
                    <ItemData ItemOID="ID.AEENDT" Value="1999-06-14" />
                    <ItemData ItemOID="ID.AESEV" Value="1" />
                    <ItemData ItemOID="ID.AEREL" Value="0" />
                    <ItemData ItemOID="ID.AEOUT" Value="1" />
                    <ItemData ItemOID="ID.AEACTTRT" Value="0" />
                    <ItemData ItemOID="ID.AECONTRT" Value="1" />
                </ItemGroupData>
                <ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="2">
                    <ItemData ItemOID="ID.TAREA" Value="ONC" />
                    <ItemData ItemOID="ID.PNO" Value="143-02" />
                    <ItemData ItemOID="ID.SCTRY" Value="USA" />
                    <ItemData ItemOID="ID.F_STATUS" Value="V" />
                    <ItemData ItemOID="ID.LINE_NO" Value="2" />
                    <ItemData ItemOID="ID.AETERM" Value="CONGESTION" />
                    <ItemData ItemOID="ID.AESTMON" Value="6" />
                    <ItemData ItemOID="ID.AESTDAY" Value="11" />
                    <ItemData ItemOID="ID.AESTYR" Value="1999" />
                    <ItemData ItemOID="ID.AESTDT" Value="1999-06-11" />
                    <ItemData ItemOID="ID.AEENMON" IsNull="Yes" />
                    <ItemData ItemOID="ID.AEENDAY" IsNull="Yes" />
                    <ItemData ItemOID="ID.AEENYR" IsNull="Yes" />
                    <ItemData ItemOID="ID.AEENDT" IsNull="Yes" />
                    <ItemData ItemOID="ID.AESEV" Value="1" />
                    <ItemData ItemOID="ID.AEREL" Value="0" />
                    <ItemData ItemOID="ID.AEOUT" Value="2" />
                    <ItemData ItemOID="ID.AEACTTRT" Value="0" />
                    <ItemData ItemOID="ID.AECONTRT" Value="1" />
                </ItemGroupData>
            </FormData>
        </StudyEventData>
    </SubjectData>
</ClinicalData>
</ODM>

```

---

## Exporting a CDISC ODM XML Document with Required and Optional Statements

### Overview

This example includes required and optional PROC CDISC statements for exporting a CDISC ODM XML document. The example illustrates how to specify metadata attributes that are stored in SAS data sets and referenced in the DATA= argument.

When exporting, the following requirements apply:

- You can specify metadata attributes for the required ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION statements either directly in the CDISC procedure statement or stored in a SAS data set.
- Metadata attributes for the optional BASICDEFINITIONS, PRESENTATION, USER, LOCATION, and SIGNATURE statements must be submitted in SAS data sets referenced in the DATA= argument.
- Optional metadata attributes for the required CLINICALDATA statement must be specified as part of the CLINICALDATA statement syntax.

### Program

First, this example uses the SAS data sets that contain metadata attributes for the required ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION statements from [“Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets”](#) on page 37.

Create additional SAS data sets that contain metadata attributes for the optional BASICDEFINITIONS, PRESENTATION, USER, LOCATION, and SIGNATURE statements. The LIBNAME statement assigns the libref CURRENT to the physical location of the SAS data sets that will store the metadata attributes.

```
libname current 'C:\mydata\';

data current.basic;
length TranslatedText $40.;
  MeasurementOID="MU.KG";
  Name="Kilogram";
  Lang="en";
  TranslatedText="English: Kilogram";
output;

  MeasurementOID="MU.KG";
  Name="Kilogram";
  Lang="sp";
  TranslatedText="Spanish: Kilogram";
output;

  MeasurementOID="MU.LB";
  Name="Pound";
  Lang="en";
```



```
        TranslatedText="English: Pound";
output;

        MeasurementOID="MU.LB";
        Name="Pound";
        Lang="sp";
        TranslatedText="Spanish: Libra";
output;
run;

data current.present;
length TranslatedText $40.;
    PresentationOID="PRES.EN";
    Lang="en";
    TranslatedText="English: Presentation";
output;

    PresentationOID="PRES.SP";
    Lang="sp";
    TranslatedText="Spanish: Presentation";
output;
run;

data current.location;
    LocationOID="LOC.CDISCHome";
    Name="CDISC Headquarters";
    Studyoid="123-456-789";
    MetadataversionOID="v1.1.0";
    EffectiveDate="2001-10-19";
    LocationType="Other";
output;

    LocationOID="LOC.site001";
    Name="Roswell Park";
    StudyOID="123-456-789";
    MetadataversionOID="v1.1.0";
    EffectiveDate="2001-10-19";
    LocationType="Site";
output;
run;

data current.user;
    length usertype $20.;
    length organization $40.;
    UserOID="USR.cdisc001";
    UserType="Other";
    FullName="Fred Flintstone";
    FirstName="Fred";
    LastName="Flintstone";
    Organization = "CDISC";
    LocationOID="LOC.CDISCHome";
    StreetName="123 Main Street";
    City="Washington";
    StateProv="DC";
    Country="United States";
    PostalCode="";
```

```

output;

    UserOID="USR.inv001";
    UserType="Investigator";
    FullName="Wilma Flintstone";
    FirstName="Wilma";
    LastName="Flintstone";
    Organization="Roswell Park";
    LocationOID="LOC.site001";
    StreetName="";
    City="";
    StateProv="";
    Country="";
    PostalCode="";
output;
run;

data current.signature;
    SignatureOID="SD.cdisc001-es";
    Methodology="Electronic";
    Meaning="Signature Meaning";
    LegalReason="LegalReason";
run;

```

Reference the SAS data sets in a PROC CDISC program:

1. The FILENAME statement assigns the fileref XMLOUT to the physical location of the output XML document.
2. The required PROC CDISC statement specifies CDISC ODM as the model and the fileref XMLOUT, which references the physical location of the output XML document.
3. The required ODM statement includes the DATA= argument to reference the SAS data set CURRENT.ODM, which stores the CDISC ODM version and file type.
4. The required STUDY statement includes the DATA= argument to reference the SAS data set CURRENT.STUDY, which stores the study identifier.
5. The required GLOBALVARIABLES statement includes the DATA= argument to reference the SAS data set CURRENT.GLOBALS, which stores general summary information about the study.
6. The optional BASICDEFINITIONS statement includes the DATA= argument to reference the SAS data set CURRENT.BASIC, which stores information about measurement units that are used in the study.
7. The required METADATAVERSION statement includes the DATA= argument to reference the SAS data set CURRENT.METADATA, which stores the metadata version and version name.
8. The optional PRESENTATION statement includes the DATA= argument to reference the SAS data set CURRENT.PRESENT, which stores information about how the study is presented to users.
9. The optional USER statement includes the DATA= argument to reference the SAS data set CURRENT.USER, which stores information about users involved in the study.

10. The optional LOCATION statement includes the DATA= argument to reference the SAS data set CURRENT.LOCATION, which stores information about the physical location of the study.
11. The optional SIGNATURE statement includes the DATA= argument to reference the SAS data set CURRENT.SIGNATURE, which stores information about the signatures required in administering the study.
12. The required CLINICALDATA statement identifies the input SAS data set, which is CURRENT.AE. The input SAS data set contains the data content and KeySet members that are written to the XML document. In addition, the CLINICALDATA statement specifies optional metadata attributes.

```

filename xmlout 'C:\XML\aeopts.xml'; 1

proc cdisc model=odm write=xmlout; 2
  odm data=current.odm; 3
  study data=current.study; 4
  globalvariables data=current.globals; 5
  basicdefinitions data=current.basic; 6
  metadataversion data=current.metadata; 7
  presentation data=current.present; 8
  user data=current.user; 9
  location data=current.location; 10
  signature data=current.signature; 11
  clinicaldata data=current.ae 12
    domain="AE"
    name = "Adverse Events"
    comment = "Adverse Events in the Clinical Trial";
run;

filename xmlout clear;

```

---

## Describing a CDISC ODM SAS Data Set with the CONTENTS Statement

### Overview

This example illustrates how to list the attributes of a CDISC ODM SAS data set in the CDISC ODM XML document named AE.XML. The output displays in the SAS log.

The example includes two different programs, which illustrate how the ODM statement LONGNAMES= processing option determines the sources of captured SAS names parameters based on the value of the option.

To view the AE.XML document, see [“Sample CDISC ODM XML Document”](#) on page 89.

### Program with LONGNAMES=NO

The following SAS program lists the attributes of a CDISC ODM SAS data set by using the ODM statement LONGNAMES=NO processing option:

1. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension).
2. The PROC CDISC statement specifies CDISC ODM as the model and the fileref XMLINP, which references the physical location of the input XML document to be imported.
3. The ODM statement specifies CDISC ODM version 1.2 and the LONGNAMES=NO processing option, which determines the following:
  - ODM name attributes are converted to SAS names that can be a maximum of eight characters in length.
  - The SAS data set name is captured from the SASDatasetName= attribute in the ODM ItemGroupDef element. SAS variable names are captured from the SASFieldName= attribute in ODM ItemDef elements. SAS format names are captured from the SASFormatName= attribute in ODM CodeList elements.
4. The CONTENTS statement writes the contents of the CDISC ODM SAS data set in the SAS log. Note that the CONTENTS statement must specify the name that is in the SASDatasetName= attribute in the ODM ItemGroupDef element.

```
filename xmlinp 'C:\XML\ae.xml'; 1

proc cdisc model=odm read=xmlinp; 2
  odm odmversion="1.2" longnames=no; 3
  contents table="AE"; 4
run;
```

**Output with LONGNAMES=NO**

Contents of CDISC ODM Table							
Name	AE						
Libref	XMLINP						
ODM Version	1.2						
AsOfDateTime	2005-05-18T14:01:41						
CreationDateTime	2005-05-18T14:01:41						
SourceSystem	SAS 9.1						
SourceSystemVersion	9.01.01M3D05172005						
Engine/Host Dependent Information							
OID	IG.AE						
Name	Adverse Events						
SASDatasetName	AE						
IsReferenceData							
Repeating	Yes						
Domain	AE						
Origin							
Role							
Purpose							
Comment	Some adverse events from this trial						
List of Variables and Attributes							
#	Variable	Datatype	Length	SigDig	Format	Label	
1	AECONTRT	string	1				
2	AEBACTTRT	string	1				
3	AEOUT	string	1				
4	AEREL	string	1				
5	AESEV	string	1				
6	AEENDT	date			DATE		
7	AEENYR	integer	4				
8	AEENDAY	integer	2				
9	AEENMON	integer	2				
10	AESTDT	date			DATE		
11	AESTYR	integer	4				
12	AESTDAY	integer	2				
13	AESTMON	integer	2				
14	AETERM	string	100				
15	LINE_NO	integer	2				
16	F_STATUS	string	1				
17	SCTRY	string	4				
18	PNO	string	15				
19	TAREA	string	4				

**Program with LONGNAMES=YES**

The following SAS program lists the attributes of a CDISC ODM SAS data set by using the ODM statement LONGNAMES=YES processing option:

1. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension).

2. The PROC CDISC statement specifies CDISC ODM as the model and the fileref XMLINP, which references the physical location of the input XML document to be imported.
3. The ODM statement specifies CDISC ODM version 1.2 and the LONGNAMES=YES processing option, which determines the following:
  - ODM name attributes are converted to SAS names that can be a maximum of 32 characters in length.
  - The SAS data set name, SAS variable names, and SAS format names are captured from the Name= attribute in the ODM ItemGroupDef, ItemDef, and CodeList elements.
4. The CONTENTS statement writes the contents of the CDISC ODM SAS data set in the SAS log. Note that the CONTENTS statement must specify the name that is in the Name= attribute in the ODM ItemGroupDef element.

```
filename xmlinp 'C:\XML\ae.xml'; 1

proc cdisc model=odm 2 read=xmlinp;
  odm odmversion="1.2" longnames=yes; 3
  contents table="Adverse_Events"; 4
run;
```

**Output with LONGNAMES=YES**

Contents of CDISC ODM Table						
Name	Adverse_Events					
Libref	XMLINP					
ODM Version	1.2					
AsOfDateTime	2005-05-18T14:01:41					
CreationDateTime	2005-05-18T14:01:41					
SourceSystem	SAS 9.1					
SourceSystemVersion	9.01.01M3D05172005					
Engine/Host Dependent Information						
OID	IG.AE					
Name	Adverse Events					
SASDatasetName	AE					
IsReferenceData						
Repeating	Yes					
Domain	AE					
Origin						
Role						
Purpose						
Comment	Some adverse events from this trial					
List of Variables and Attributes						
#	Variable	Datatype	Length	SigDig	Format	Label
1	Actions_taken_other	string	1			
2	Actions_taken_re_study_drug	string	1			
3	Outcome	string	1			
4	Relationship_to_study_drug	string	1			
5	Severity	string	1			
6	Derived_Stop_Date	date			DATE	
7	Stop_Year__Enter_Four_Digit_Yea	integer	4			
8	Stop_Day__Enter_Two_Digits_01_3	integer	2			
9	Stop_Month__Enter_Two_Digits_01	integer	2			
10	Derived_Start_Date	date			DATE	
11	Start_Year__Enter_Four_Digit_Ye	integer	4			
12	Start_Day__Enter_Two_Digits_01_	integer	2			
13	Start_Month__Enter_Two_Digits_0	integer	2			
14	Conmed_Indication	string	100			
15	Line_Number	integer	2			
16	Record_status__5_levels__interna	string	1			
17	Country	string	4			
18	Protocol_Number	string	15			
19	Therapeutic_Area	string	4			

---

## Listing a Directory with the DATASETS Statement

### Overview

This example illustrates how to produce a directory listing in the SAS log of a CDISC ODM XML document. The output lists the name of the CDISC ODM SAS data set that is in the XML document named AE.XML. The output also lists the attributes of the XML document, such as ODM version and date of creation.

The example includes two different programs, which illustrate how the ODM statement `LONGNAMES=` processing option determines the sources of captured SAS names based on the value of the option.

To view the AE.XML document, see [“Sample CDISC ODM XML Document”](#) on page 89.

### Program with `LONGNAMES=NO`

The following SAS program produces a directory listing by using the ODM statement `LONGNAMES=NO` processing option:

1. The `FILENAME` statement assigns the fileref `XMLINP` to the physical location of the input XML document (complete pathname, filename, and file extension).
2. The `PROC CDISC` statement specifies CDISC ODM as the model and the fileref `XMLINP`, which references the physical location of the input XML document to be imported.
3. The ODM statement specifies CDISC ODM version 1.2 and the `LONGNAMES=NO` processing option, which determines the following:
  - ODM name attributes are converted to SAS names that can be a maximum of eight characters in length.
  - The SAS data set name is captured from the `SASDatasetName=` attribute in the ODM `ItemGroupDef` element.
4. The `DATASETS` statement produces a directory listing of the XML document in the SAS log.

```
filename xmlinp 'C:\XML\ae.xml'; 1

proc cdisc model=odm read=xmlinp; 2
  odm odmversion="1.2" longnames=no; 3
  datasets; 4
run;
```



**Output with LONGNAMES=NO****Output 4.3** Directory Listing of the AE.XML Document in the SAS Log

```

                                Directory of CDISC ODM Tables

Libref          XMLINP
ODM Version     1.2
AsOfDateTime    2005-05-18T14:01:41
CreationDateTime 2005-05-18T14:01:41
SourceSystem    SAS 9.1
SourceSystemVersion 9.01.01M3D05172005

                                #  Tablename

                                1  AE

```

**Program with LONGNAMES=YES**

The following SAS program produces a directory listing by using the ODM statement LONGNAMES=YES processing option:

1. The FILENAME statement assigns the fileref XMLINP to the physical location of the input XML document (complete pathname, filename, and file extension).
2. The PROC CDISC statement specifies CDISC ODM as the model and the fileref XMLINP, which references the physical location of the input XML document to be imported.
3. The ODM statement specifies CDISC ODM version 1.2 and the LONGNAMES=YES processing option, which determines the following:
  - ODM name attributes are converted to SAS names that can be a maximum of 32 characters in length.
  - The SAS data set name is captured from the Name= attribute in the ODM ItemGroupDef element.
4. The DATASETS statement produces a directory listing of the XML document in the SAS log.

```

filename xmlinp 'C:\XML\ae.xml'; 1

proc cdisc model=odm read=xmlinp; 2
  odm odmversion="1.2" longnames=yes; 3
  datasets; 4
run;

```

### Output with LONGNAMES=YES

#### Output 4.4 Directory Listing of the AE.XML Document in the SAS Log

```

                                     Directory of CDISC ODM Tables
Libref          XMLINP
ODM Version     1.2
AsOfDateTime    2005-05-18T14:01:41
CreationDateTime 2005-05-18T14:01:41
SourceSystem    SAS 9.1
SourceSystemVersion 9.01.01M3D05172005

                                     # Tablename
                                     1 Adverse_Events
```

## Chapter 5

# CDISC Procedure Examples for CDISC SDTM

<b>Validating CDISC SDTM Data in a SAS Data Set</b> . . . . .	<b>53</b>
Overview . . . . .	53
Program . . . . .	53
Output . . . . .	54
<b>Validating CDISC SDTM Data in an Oracle Table</b> . . . . .	<b>54</b>
Overview . . . . .	54
Program . . . . .	54

## Validating CDISC SDTM Data in a SAS Data Set

### Overview

This example validates CDISC SDTM data that is stored in a SAS data set.

### Program

The following SAS program validates CDISC SDTM data that is stored in the SAS data set RESULTS.AE:

1. The LIBNAME statement assigns the libref RESULTS to the physical location of the input SAS data set to be validated.
2. The PROC CDISC statement specifies CDISC SDTM as the model.
3. The SDTM statement specifies the SDTM version number.
4. The DOMAINDATA statement specifies the SAS data set to be validated, the two-character domain code, and the domain model type.

```
libname results 'C:\Myfiles\'; 1

proc cdisc model=sdm; 2
  sdm sdmversion="3.1"; 3
  domaindata data=results.AE domain=ae category=events; 4
run;

libname results clear;
```

## Output

### Log 5.1 SAS Log Output

```
21 proc cdisc model=sdm;  
22     sdm sdmversion="3.1";  
23     domaindata data=results.ae domain=ae category=events;  
24 run;  
  
ERROR: Required parameters not contained on DOMAINDATA(Domain=AE) statement.  
       Required parameter STUDYID not present.  
       Required parameter DOMAIN not present.  
       Required parameter USUBJID not present.  
       Required parameter AESEQ not present.  
       Required parameter AEDECOD not present.  
NOTE: The SAS System stopped processing this step because of errors.  
NOTE: PROCEDURE CDISC used (Total process time):  
       real time          0.00 seconds  
       cpu time           0.00 seconds
```

---

## Validating CDISC SDTM Data in an Oracle Table

### Overview

This example validates CDISC SDTM data that is stored in an Oracle table.

### Program

The following SAS program validates CDISC SDTM data that is stored in an Oracle table named AE:

1. The LIBNAME statement for the SAS/ACCESS Interface to Oracle engine assigns the libref ORALIB to the physical location of the Oracle database that contains the Oracle table.
2. The PROC CDISC statement specifies CDISC SDTM as the model.
3. The SDTM statement specifies the SDTM version number.
4. The DOMAINDATA statement specifies the Oracle table to be validated, the two-character domain code, and the domain model type.

```
libname oralib oracle user=myuser pw=mypw
  path=ora_dbms preserve_tab_names=yes
  connection=sharedread schema=myschema; 1

proc cdisc model=sdtm; 2
  sdtm sdtmversion="3.1"; 3
  domaindata data=oralib.AE domain=ae category=events; 4
run;

libname oralib clear;
```



## Part 3

---

# Procedure Reference

<i>Chapter 6</i>	
<b>CDISC ODM Procedure</b> .....	59
<i>Chapter 7</i>	
<b>CDISC SDTM Procedure</b> .....	83





## Chapter 6

# CDISC ODM Procedure

---

<b>Overview: CDISC ODM Procedure</b> . . . . .	<b>59</b>
<b>Syntax: CDISC ODM Procedure</b> . . . . .	<b>60</b>
PROC CDISC for ODM Statement . . . . .	61
ODM Statement . . . . .	63
STUDY Statement . . . . .	70
GLOBALVARIABLES Statement . . . . .	71
BASICDEFINITIONS Statement . . . . .	71
METADATAVERSION Statement . . . . .	72
PRESENTATION Statement . . . . .	73
USER Statement . . . . .	74
LOCATION Statement . . . . .	76
SIGNATURE Statement . . . . .	77
CLINICALDATA Statement . . . . .	78
CONTENTS Statement . . . . .	80
DATASETS Statement . . . . .	80

---

## Overview: CDISC ODM Procedure

PROC CDISC imports and exports XML documents that conform to CDISC ODM version 1.2.

The following table provides a quick reference that indicates the required and optional statements for importing and exporting a CDISC ODM XML document:

**Table 6.1** Table of Procedure Tasks

Statement	Task	Importing	Exporting
<a href="#">PROC CDISC for ODM Statement on page 61</a>	Specifies CDISC ODM as the model and references the physical location of the input or output XML document	required	required
<a href="#">ODM Statement on page 63</a>	Specifies the CDISC ODM version number and file type	required	required
<a href="#">STUDY Statement on page 70</a>	Specifies the study identifier	not valid	required

Statement	Task	Importing	Exporting
<a href="#">GLOBALVARIABLES Statement on page 71</a>	Specifies general summary information about the study	not valid	required
<a href="#">BASICDEFINITIONS Statement on page 71</a>	Specifies information about measurement units that are used in the study	not valid	optional
<a href="#">METADATAVERSION Statement on page 72</a>	Specifies the metadata version and version name that are used by the study	not valid	required
<a href="#">PRESENTATION Statement on page 73</a>	Specifies information about how the study is presented to users	not valid	optional
<a href="#">USER Statement on page 74</a>	Specifies information about users involved in the study	not valid	optional
<a href="#">LOCATION Statement on page 76</a>	Specifies information about the physical location of the study	not valid	optional
<a href="#">SIGNATURE Statement on page 77</a>	Specifies information about the signatures required in administering the study	not valid	optional
<a href="#">CLINICALDATA Statement on page 78</a>	When importing, identifies the output SAS data set and specifies where the data content in the XML document begins; when exporting, identifies the input SAS data set and optional metadata attributes	required	required
<a href="#">CONTENTS Statement on page 80</a>	Describes the contents of a CDISC ODM SAS data set in the SAS log	optional	not valid
<a href="#">DATASETS Statement on page 80</a>	Produces a directory listing of the CDISC ODM XML document in the SAS log	optional	not valid

---

## Syntax: CDISC ODM Procedure

**Restrictions:** PROC CDISC is supported in the following operating environments: Windows, UNIX, and z/OS.  
 PROC CDISC supports only one CDISC model and one CLINICALDATA statement per invocation.  
 PROC CDISC does not support vendor extensions to CDISC ODM.

**Tip:** PROC CDISC statements are listed in CDISC ODM markup order.

---

**PROC CDISC MODEL=ODM**

```

READ=fileref | WRITE=fileref
  <FORMATACTIVE= NO | YES >
  <FORMATNOREPLACE= NO | YES>
  <FORMATLIBRARY=libref>
  <LANGUAGE=language-identifier> ;
ODM <processing-options> metadata-attributes |
  <processing-options> DATA=libref.member-name ;
STUDY metadata-attributes | DATA=libref.member-name ;
GLOBALVARIABLES metadata-attributes | DATA=libref.member-name ;
  <BASICDEFINITIONSDATA=libref.member-name> ;
METADATAVERSION metadata-attributes | DATA=libref.member-name ;
  <PRESENTATIONDATA=libref.member-name> ;
  <USERDATA=libref.member-name> ;
  <LOCATIONDATA=libref.member-name> ;
  <SIGNATUREDATA=libref.member-name> ;
CLINICALDATA
  OUT=libref.member-name SASDATASETNAME="name" |
  DATA=libref.member-name <metadata-attributes> ;
CONTENTS TABLE="name";
DATASETS;

```

---

**PROC CDISC for ODM Statement**

Specifies CDISC ODM as the model and references the physical location of the input or output ODM XML document.

**Requirement:** Required for both importing and exporting.

---

**Syntax****PROC CDISC MODEL=ODM**

```

READ=fileref | WRITE=fileref
  <FORMATACTIVE= NO | YES >
  <FORMATNOREPLACE= NO | YES>
  <FORMATLIBRARY=libref>
  <LANGUAGE=language-identifier> ;

```

**Summary of Optional Arguments**

```

FORMATACTIVE=NO | YES
FORMATLIBRARY=libref
FORMATNOREPLACE=NO | YES
LANGUAGE="language-identifier"
MODEL=ODM
READ=fileref
WRITE=fileref

```

## Arguments

### MODEL=ODM

specifies CDISC ODM as the model.

**Requirement:** Required for both importing and exporting.

**Interaction:** The model specification determines the syntax for the PROC CDISC statement and subsequent procedure statements. Specifying MODEL=ODM requires that you specify either the READ= or WRITE= argument and that the next statement must be the ODM statement.

### READ=*fileref*

specifies the SAS fileref that is assigned to the input XML document.

**Requirement:** Required for importing.

### WRITE=*fileref*

specifies the SAS fileref that is assigned to the output XML document.

**Requirement:** Required for exporting.

### FORMATACTIVE=NO | YES

specifies whether the content in the CDISC ODM CodeList element, which includes instructions for transcoding display data in an XML document, is to be converted to PROC FORMAT style formats or SAS formats, and vice versa.

For both importing and exporting, FORMATACTIVE=NO causes formats to be ignored. This is the default.

For importing, FORMATACTIVE=YES does the following:

- Converts CDISC ODM CodeList content to the corresponding SAS format, or creates PROC FORMAT style formats from the CodeList content.
- Registers the SAS formats in the referenced variables.
- Stores the created SAS formats in the SAS FORMAT library.

For exporting, FORMATACTIVE=YES converts SAS formats to the corresponding CDISC ODM CodeList content and variable references.

**Default:** NO

**Requirement:** Optional for both importing and exporting.

**Interaction:** (Optional) If you specify FORMATACTIVE=YES, then you can specify FORMATNOREPLACE= and FORMATLIBRARY=.

### FORMATNOREPLACE=NO | YES

specifies whether to replace existing SAS formats in the FORMAT catalog when an existing SAS format has the same name as a format that is being converted by PROC CDISC.

When FORMATNOREPLACE=NO, the procedure replaces existing SAS formats that have the same name. This is the default.

When FORMATNOREPLACE=YES, the procedure does not replace existing SAS formats that have the same name.

**Default:** NO

**Requirement:** Optional for importing.

**Interaction:** Ignored unless FORMATACTIVE=YES.

### FORMATLIBRARY=*libref*

specifies the assigned libref of an existing SAS library in which to create the FORMAT catalog, which is a permanent storage location for variable formats that are created by PROC CDISC.

**Requirement:** Optional for importing.

**Interaction:** Ignored unless FORMATACTIVE=YES.

**LANGUAGE="language-identifier"**

specifies a language identifier for the language tag attribute (xml:lang) in the ODM TranslatedText elements. Enclose the language identifier in single or double quotation marks.

- When importing, PROC CDISC locates the specified language identifier in the ODM TranslatedText element, and creates a SAS format by using the TranslatedText items with a matching language tag attribute. The created SAS format is then applied to the data that is imported from the XML document.
- When exporting, PROC CDISC uses the specified language identifier as the language tag attribute value in the ODM TranslatedText element.

A language identifier, as defined in the XML specification, can be one of the following:

- A two-letter language code as defined by ISO 639, "Codes for the representation of names of languages."
- A language identifier that has been registered with the Internet Assigned Numbers Authority (IANA). The language identifier begins with the prefix i- or I-.
- A language identifier that has been assigned by the user or agreed on between parties in private use. The language identifier begins with the prefix x- or X- to ensure that it does not conflict with names that are later standardized or registered with IANA.

**Requirement:** Optional for both importing and exporting.

**See:** [“Importing a CDISC ODM XML Document Using a Language Identifier” on page 22](#)

---

## ODM Statement

Specifies the CDISC ODM version number and file type.

**Restriction:** Processing options must be specified in the statement. They cannot be stored in a SAS data set. Only metadata attributes can be stored in a SAS data set.

**Requirement:** Required for both importing and exporting.

**Tip:** You can specify metadata attributes either directly in the ODM statement or store them in a SAS data set that you reference in the DATA= argument.

---

## Syntax

**ODM** <processing-options> metadata-attributes | <processing-options> **DATA**=libref.member-name ;

## Summary of Optional Arguments

*DATA=libref.member-name*

*metadata-attributes*

*processing-options*

## Arguments

### *processing-options*

specifies options that affect how KeySet members are processed.

LONGNAMES=NO | YES

determines the sources of captured SAS name parameters, and controls the maximum length of SAS names that is valid.

NO

When importing, specifies that ODM name attributes are converted to SAS names that can be a maximum length of eight characters. PROC CDISC captures the SAS data set name from the SASDatasetName= attribute in the ODM ItemGroupDef element. SAS variable names are captured from the SASFieldName= or SDSVarName= attribute in ODM ItemDef elements. SAS format names are captured from the SASFormatName= attribute in ODM CodeList elements. If these optional ODM attributes are not available, then the Name= attribute (which is required in the individual ODM elements) is used.

When exporting, specifies that SAS names are converted to ODM name attributes that can be a maximum length of eight characters. PROC CDISC exports the SAS data set name to the SASDatasetName= attribute in the ODM ItemGroupDef element. SAS variable names are exported to the SASFieldName= in ODM ItemDef elements. SAS format names are exported to the SASFormatName= attribute in ODM CodeList elements. The required ODM Name= attribute is a duplicate of the SAS name attribute value.

For example, by using the [“Sample CDISC ODM XML Document”](#) on page 89, LONGNAMES=NO imports the SAS data set name AE from the SASDatasetName= attribute in the ODM ItemGroupDef element.

**Requirement:** When importing, values for the ODM elements must be valid SAS names. Except for in the Name= attribute, an invalid SAS name generates an error because it does not conform to the ODM schema.

**Interaction:** PROC CDISC behavior with LONGNAMES=NO is compatible with legacy data before Version 7, which includes Version 5 transport files.

YES

When importing, specifies that ODM name attributes are converted to SAS names that can be a maximum length of 32 characters. PROC CDISC captures the SAS data set name, SAS variable names, and SAS format names from the Name= attribute in the ODM ItemGroupDef, ItemDef, and CodeList elements. The optional SASDatasetName=, SASFieldName=, SDSVarName=, and SASFormatName= attributes (used with LONGNAMES=NO) are ignored even if they are available in the ODM elements.

When exporting, specifies that SAS names are converted to ODM name attributes that can be a maximum length of 32 characters. PROC CDISC exports the SAS data set name, SAS variable names, and SAS format names as values for the Name= attribute in the ODM ItemGroupDef, ItemDef, and CodeList elements.

For example, by using the [“Sample CDISC ODM XML Document”](#) on page 89, LONGNAMES=YES imports the SAS data set name Adverse\_Events from the Name= attribute in the ODM ItemGroupDef element.

**Interactions:**

PROC CDISC behavior with LONGNAMES=YES is compatible with Version 7 and later. The behavior is not compatible with Version 5 transport files.

When importing, if a Name= attribute value is not a valid SAS name, PROC CDISC converts it to a valid SAS name. For example, a blank space or any other invalid character in the field is replaced with an underscore. Truncation occurs if the Name= attribute value exceeds the maximum length.

**Default:** NO

**Requirement:** Optional for both importing and exporting.

**Tip:** The SASDatasetName=, SASFieldName= or SDSVarName=, and SASFormatName= ODM elements are optional in CDISC ODM version 1.2. However, the Name= attribute for those ODM elements is required.

**See:** [“CDISC ODM KeySet Members” on page 6](#)

ODMMAXIMUMOIDLENGTH=*number*

specifies a character length for the CDISC ODM KeySet members. The default value is the OID length that is defined in CDISC ODM. The maximum OID length that PROC CDISC accepts is 100 characters.

**Requirement:** Optional for importing.

**See:**

[“CDISC ODM KeySet Members” on page 6](#)

[“Importing a CDISC ODM XML Document Specifying KeySet Processing Options” on page 20](#)

ODMMINIMUMKEYSET=NO | YES

specifies whether to limit the CDISC ODM KeySet members that are in the study data.

NO

When importing, specifies that all KeySet members are written to the output SAS data set. Note that ODMMINIMUMKEYSET=NO increases the row size 10 times the maximum OID length that is defined in CDISC ODM.

When exporting, specifies that all KeySet members that are in the input SAS data set are written to the output XML document.

YES

When importing, specifies that only the SubjectKey is written to the output SAS data set.

When exporting, specifies that only the SubjectKey that is in the input SAS data set is written to the output XML document. Other KeySet members are from PROC CDISC statements or are automatically generated by PROC CDISC.

**Default:** NO

**Requirement:** Optional for both importing and exporting.

**See:**

[“CDISC ODM KeySet Members” on page 6](#)

[“Importing a CDISC ODM XML Document Using Default KeySet Processing” on page 18](#) and [“Importing a CDISC ODM XML Document Specifying KeySet Processing Options” on page 20](#)

**ORDERNUMBER=YES | NO**

determines whether PROC CDISC validates OrderNumber attributes in ItemRef elements. OrderNumber attributes define an order among related entities by using consecutive integer values.

**YES**

specifies to validate OrderNumber attributes. If an element contains OrderNumber attributes that do not conform to CDISC ODM, such as missing or out-of-sequence integers, PROC CDISC displays warnings in the SAS log.

**Alias:** USE

**NO**

specifies to ignore OrderNumber attributes. Warnings are not displayed in the SAS log.

**Alias:** IGNORE

**Default:** YES

**Requirement:** Optional for importing.

**See:** [“Importing a CDISC ODM XML Document with OrderNumber Attributes” on page 28](#)

**USENAMEASLABEL=NO | YES**

determines the sources of captured SAS data set and variable labels. Labels can be a maximum of 200 characters.

**NO**

When importing, specifies that ODM comment attributes are converted to SAS labels. PROC CDISC imports the SAS data set label from the Comment= attribute in the ODM ItemGroupDef element, and SAS variable labels from the Comment= attribute in the ODM ItemDef elements.

When exporting, specifies that SAS labels are converted to ODM comment attributes. PROC CDISC exports the SAS data set label to the Comment= attribute in the ODM ItemGroupDef element, and SAS variable labels to the Comment= attribute in the ODM ItemDef elements.

**YES**

When importing, specifies that ODM name attributes are converted to SAS labels. The value of the LONGNAMES= option determines the sources of the captured SAS data set and variable labels:

**LONGNAMES=NO**

PROC CDISC imports the SAS data set label from the SASDatasetName= attribute in the ODM ItemGroupDef element, and SAS variable labels from the SASFieldName= or SDSVarName= attribute in ODM ItemDef elements. If these optional ODM attributes are not available, then the Name= attribute (which is required in the individual ODM elements) is used.

**LONGNAMES=YES**

PROC CDISC imports the SAS data set label from the Comment= attribute in the ODM ItemGroupDef element, and SAS variable labels from the Comment= attribute in the ODM ItemDef elements.

**Restriction:** Specifying USENAMEASLABEL=YES and LONGNAMES=YES are mutually exclusive.

When exporting, specifies that SAS labels are converted to ODM name attributes. The value of the LONGNAMES= option determines the sources of the captured SAS data set and variable labels:



**LONGNAMES=NO**

PROC CDISC exports the SAS data set label to the SASDatasetName= attribute in the ODM ItemGroupDef element, and SAS variable labels to the SASFieldName= in ODM ItemDef elements. The required ODM Name= attribute is a duplicate of the SAS name attribute value.

**LONGNAMES=YES**

PROC CDISC exports the SAS data set label to the Comment= attribute in the ODM ItemGroupDef element, and SAS variable labels to the Comment= attribute in the ODM ItemDef elements.

**Restriction:** Specifying USENAMEASLABEL=YES and LONGNAMES=YES are mutually exclusive.

**Default:** NO

**Requirement:** Optional for both importing and exporting.

**Tip:** The Comment= attribute in the ODM ItemGroupDef and ItemDef elements is optional in CDISC ODM version 1.2. However, the Name= attribute for those ODM elements is required.

**metadata-attributes**

includes the following syntax, which can be specified directly in the ODM statement or stored in a SAS data set that you reference in the DATA= argument:

**ODMVERSION="version-number"**

specifies the CDISC ODM version number. The valid value is 1.2. Enclose the version number in single or double quotation marks.

**Restriction:** PROC CDISC does not support CDISC ODM versions before 1.2. CDISC ODM versions 1.1 and 1.0 are not supported. However, CDISC ODM version 1.2.1 is semantically equivalent to version 1.2 and should be specified as simply 1.2.

**Requirement:** Required for both importing and exporting.

**FILEOID="identifier"**

specifies a unique identifier for the exported XML document. Enclose the identifier in single or double quotation marks.

**Requirement:** Required for exporting.

**FILETYPE=SNAPSHOT | TRANSACTIONAL**

specifies the XML document file type, which defines the type of data that the XML document contains.

**SNAPSHOT**

specifies an XML document that contains only the current state of the data and metadata that it describes, with no transactional history.

**TRANSACTIONAL**

specifies an XML document that contains the current state of the data and metadata, with transactional history. A transactional XML document supports more than one instruction per data point.

**Requirement:** The input SAS data set that you are exporting must contain the required data. If you specify TRANSACTIONAL, the transactional history must be in the SAS data set.

**Requirement:** Required for exporting.

**DESCRIPTION="string"**

specifies a text string that provides details to supplement the other attributes that are described in the XML document. Enclose the string in single or double quotation marks.

**Requirement:** Optional for exporting.

GRANULARITY=ALL | METADATA | ADMINDATA | REFERENCEDATA | ALLCLINICALDATA | SINGLESITE | SINGLESUBJECT  
describes the scope of information in the XML document.

ALL

indicates any and all types of data and metadata.

METADATA

indicates only metadata.

ADMINDATA

indicates the AdminData element, which is administrative data about users, locations, and electronic signatures.

REFERENCEDATA

indicates the ReferenceData element, which is reference data that provides information about how to interpret clinical data.

ALLCLINICALDATA

indicates all clinical data.

SINGLESITE

indicates clinical data for a single site.

SINGLESUBJECT

indicates clinical data for a single subject.

**Requirement:** Optional for exporting.

**Interaction:** PROC CDISC does not generate output that is based on the specified value of GRANULARITY=.

ARCHIVAL=YES

when FILETYPE=TRANSACTIONAL, indicates whether the XML document should meet the requirements of an electronic record as defined in the FDA 21 CFR Part 11 standard.

**Requirement:** Optional for exporting.

CREATIONDATETIME="*datetime-value*"

specifies the date and time when the XML document was created or transmitted, in compliance with ISO 8601 guidelines. Enclose the value in single or double quotation marks.

**Requirement:** Optional for exporting.

PRIORFILEOID="*name*"

specifies a reference to the previous XML document (if any) in a series. Enclose the name in single or double quotation marks.

**Requirement:** Optional for exporting.

ASOFDATETIME="*datetime-value*"

specifies the date and time when the source database was queried to create the XML document, in compliance with ISO 8601 guidelines. Enclose the value in single or double quotation marks.

**Requirement:** Optional for exporting.

ORIGINATOR="*name*"

identifies the organization that generated the XML document. Enclose the name in single or double quotation marks.

**Requirement:** Optional for exporting.

**SOURCESYSTEM="string"**

specifies the application that created or transmitted the XML document. The default value is the short name of the current SAS release (for example, **SAS 9.3**). Enclose the string in single or double quotation marks.

**Requirement:** Optional for exporting.

**SOURCESYSTEMVERSION="string"**

specifies the version of the application that created or transmitted the XML document. The default value is the annotated name of the SAS release (for example, **9.01.01MxPmmddyyy**). Enclose the string in single or double quotation marks.

**Requirement:** Optional for exporting.

**Restriction:** You cannot store some metadata attributes in a SAS data set, and also specify other metadata attributes in the ODM statement. If you use a SAS data set, you must store all metadata attributes in the SAS data set.

**DATA=libref.member-name**

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

## Details

The following table lists processing options and CDISC ODM metadata attributes that are available for importing or exporting, and whether they can be stored in a SAS data set.

**Table 6.2** ODM Statement Syntax Quick Reference

Syntax	Importing	Exporting	Stored in SAS Data Set
LONGNAMES=NO   YES	optional	optional	no
ODMMAXIMUMOIDLENGTH=	optional	not valid	no
ODMMINIMUMKEYSET=NO   YES	optional	optional	no
ORDERNUMBER=YES   NO	optional	not valid	no
USERNAMEASLABEL=NO   YES	optional	optional	no
ODMVERSION=	required	required	yes
FILEOID=	not valid	required	yes
FILETYPE=	not valid	required	yes
DESCRIPTION=	not valid	optional	yes
GRANULARITY=	not valid	optional	yes
ARCHIVAL=YES	not valid	optional	yes

Syntax	Importing	Exporting	Stored in SAS Data Set
CREATIONDATETIME=	not valid	optional	yes
PRIORFILEOID=	not valid	optional	yes
ASOFDATETIME=	not valid	optional	yes
ORIGINATOR=	not valid	optional	yes
SOURCESYSTEM=	not valid	optional	yes
SOURCESYSTEMVERSION=	not valid	optional	yes

---

## STUDY Statement

Specifies the study identifier.

**Requirement:** Required for exporting.

**Tip:** You can specify the metadata attribute either directly in the STUDY statement or store it in a SAS data set that you reference in the DATA= argument.

---

### Syntax

**STUDY** *metadata-attribute* | **DATA=***libref.member-name* ;

### Summary of Optional Arguments

*DATA=libref.member-name*  
*metadata-attribute*

### Arguments

#### *metadata-attribute*

includes the following syntax, which can be specified directly in the STUDY statement or stored in a SAS data set that you reference in the DATA= argument:

**STUDYOID=***"ODM-identifier"*

specifies a unique identifier for the study, which overrides the `__STUDYOID` variable in the input SAS data set. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

#### **DATA=***libref.member-name*

specifies the SAS data set that contains the metadata attribute. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

---

## GLOBALVARIABLES Statement

Specifies general summary information about the study.

**Requirement:** Required for exporting.

**Tip:** You can specify metadata attributes either directly in the GLOBALVARIABLES statement or store them in a SAS data set that you reference in the DATA= argument.

---

### Syntax

**GLOBALVARIABLES** *metadata-attributes* | **DATA=***libref.member-name* ;

### Required Arguments

#### *metadata-attributes*

includes the following syntax, which can be specified directly in the GLOBALVARIABLES statement or stored in a SAS data set that you reference in the DATA= argument:

STUDYNAME="*name*"

specifies the short external name of the study. Enclose the name in single or double quotation marks.

**Requirement:** Required.

STUDYDESCRIPTION="*string*"

specifies a description of the study. Enclose the string in single or double quotation marks.

**Requirement:** Required.

PROTOCOLNAME="*name*"

specifies the sponsor's internal name for the protocol. Enclose the name in single or double quotation marks.

**Requirement:** Required

**Restriction:** You cannot store some metadata attributes in a SAS data set, and also specify other metadata attributes in the GLOBALVARIABLES statement. If you use a SAS data set, you must store all metadata attributes in the SAS data set.

#### **DATA=***libref.member-name*

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

---

## BASICDEFINITIONS Statement

Specifies information about measurement units that are used in the study.

**Requirements:** Optional for exporting.

Metadata attributes must be stored in a SAS data set that you reference in the DATA= argument. You cannot specify metadata attributes in the BASICDEFINITIONS statement.

---

## Syntax

**BASICDEFINITIONS DATA**=*libref.member-name* ;

### Required Argument

**DATA**=*libref.member-name*

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

**MEASUREMENTOID**="*ODM-identifier*"

specifies a symbol or abbreviation that represents a measurement unit. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

**NAME**="*name*"

specifies the name of the measurement unit. Enclose the name in single or double quotation marks.

**Requirement:** Required.

**LANGUAGE**="*language-identifier*"

specifies a language identifier. Enclose the identifier in single or double quotation marks.

A language identifier, as defined in the XML specification, can be one of the following:

- A two-letter language code as defined by ISO 639, “Codes for the representation of names of languages.”
- A language identifier that has been registered with the IANA. The language identifier begins with the prefix i- or I-.
- A language identifier that has been assigned by the user or agreed on between parties in private use. The language identifier must begin with the prefix x- or X- to ensure that it does not conflict with names that are later standardized or registered with IAN.

**Requirement:** Required.

**TRANSLATEDTEXT**="*string*"

specifies the name of the measurement unit in the specified language. Enclose the string in double or single quotation marks.

**Requirement:** Required.

---

## METADATAVERSION Statement

Specifies the metadata version and version name that are used by the study.

**Requirement:** Required for exporting.

**Tip:** You can specify metadata attributes either directly in the METADATAVERSION statement or store them in a SAS data set that you reference in the DATA= argument.

---

## Syntax

**METADATAVERSION** *metadata-attributes* | **DATA=***libref.member-name* ;

### Required Arguments

#### *metadata-attributes*

includes the following syntax, which can be specified directly in the METADATAVERSION statement or stored in a SAS data set that you reference in the DATA= argument:

**METADATAVERSIONOID=***"name"*

specifies the metadata version that is used by the study. Enclose the name in single or double quotation marks.

**Requirement:** Required.

**NAME=***"name"*

specifies a name for the metadata version. Enclose the name in single or double quotation marks.

**Requirement:** Required

**Restriction:** You cannot store some metadata attributes in a SAS data set, and also specify other metadata attributes in the METADATAVERSION statement. If you use a SAS data set, you must store all metadata attributes in the SAS data set.

#### **DATA=***libref.member-name*

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

---

## PRESENTATION Statement

Specifies information about how the study is presented to users.

**Requirements:** Optional for exporting.

Metadata attributes must be stored in a SAS data set that you reference in the DATA= argument. You cannot specify metadata attributes in the PRESENTATION statement.

---

## Syntax

**PRESENTATION** **DATA=***libref.member-name* ;

### Required Argument

#### **DATA=***libref.member-name*

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

**PRESENTATIONOID=***"ODM-identifier"*

specifies a reference to a presentation definition. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

LANGUAGE="*language-identifier*"

specifies a language identifier. Enclose the identifier in single or double quotation marks.

A language identifier, as defined in the XML specification, can be one of the following:

- A two-letter language code as defined by ISO 639, “Codes for the representation of names of languages.”
- A language identifier that has been registered with the IANA. The language identifier begins with the prefix i- or I-.
- A language identifier that has been assigned by the user or agreed on between parties in private use. The language identifier must begin with the prefix x- or X- to ensure that it does not conflict with names that are later standardized or registered with IANA.

**Requirement:** Required.

TRANSLATEDTEXT="*string*"

specifies a portion of the presentation in the specified language. Enclose the string in single or double quotation marks.

**Requirement:** Required.

---

## USER Statement

Specifies information about users involved in the study.

**Restriction:** This is an advanced statement. You must be familiar with CDISC ODM.

**Requirements:** Optional for exporting.

Metadata attributes must be stored in a SAS data set that you reference in the DATA= argument. You cannot specify metadata attributes in the USER statement.

---

## Syntax

USER DATA=*libref.member-name* ;

### Required Argument

DATA=*libref.member-name*

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

USEROID="*ODM-identifier*"

specifies a reference to a user. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

USERTYPE=SPONSOR | INVESTIGATOR | LAB | OTHER

specifies the user's role in the study.

**Requirement:** Optional.



LOGINNAME="*user-ID*"

specifies the user ID that the user uses to log in to the clinical trials data system. Enclose the user ID in single or double quotation marks.

**Requirement:** Optional.

DISPLAYNAME="*name*"

specifies a short name for the user. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

FULLNAME="*name* "

specifies the full name of the user. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

LASTNAME="*name* "

specifies the last name of the user. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

FIRSTNAME="*name* "

specifies the first name of the user. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

ORGANIZATION="*name* "

specifies the user's organization. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

STREETNAME="*street-address*"

specifies the street address in the user's postal address. Enclose the address in single or double quotation marks.

**Requirement:** Optional.

CITY="*name* "

specifies the city name in the user's postal address. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

STATEPROV="*state-or-province*"

specifies the state or province in the user's postal address. Enclose the state or province in single or double quotation marks.

**Requirement:** Optional.

COUNTRY="*name* "

specifies the country name in the user's postal address. This value must be an ISO 3166 two-letter country code. Enclose the name in single or double quotation marks.

**Requirement:** Optional.

POSTALCODE="*code* "

specifies the postal code in the user's postal address. Enclose the code in single or double quotation marks.

**Requirement:** Optional.

OTHERTEXT="*string*"

specifies any other text that is needed in the user's postal address. Enclose the text in single or double quotation marks.

**Requirement:** Optional.

EMAIL="*e-mail-address*"

specifies the user's e-mail address. Enclose the address in single or double quotation marks.

**Requirement:** Optional.

PICTFILENAME="*filename* "

specifies a filename that contains a picture of the user. Enclose the filename in single or double quotation marks.

**Requirement:** Optional.

PICTIMAGETYPE="*file-type* "

specifies the image file type. Enclose the file type in single or double quotation marks.

**Requirement:** Optional.

PAGER="*number*"

specifies the user's pager number. Enclose the number in single or double quotation marks.

**Requirement:** Optional.

FAX="*number*"

specifies the user's fax number. Enclose the number in single or double quotation marks.

**Requirement:** Optional.

PHONE="*number* "

specifies the user's phone number. Enclose the number in single or double quotation marks.

**Requirement:** Optional.

LOCATIONOID="*ODM-identifier*"

specifies a reference to a location definition. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Optional.

---

## LOCATION Statement

Specifies information about the physical location of the study.

**Restriction:** This is an advanced statement. You must be familiar with CDISC ODM.

**Requirements:** Optional for exporting.

Metadata attributes must be stored in a SAS data set that you reference in the DATA= argument. You cannot specify metadata attributes in the LOCATION statement.

---

## Syntax

LOCATION DATA=*libref.member-name* ;

## Required Argument

### DATA=*libref.member-name*

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

### LOCATIONOID="*ODM-identifier*"

specifies a unique identifier for a location. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

### NAME="*location-name*"

specifies the name of the location. Enclose the name in single or double quotation marks.

**Requirement:** Required.

### LOCATIONTYPE=SPONSOR | SITE | CRO | LAB | OTHER

specifies the type of location.

**Requirement:** Required.

### STUDYOID="*ODM-identifier*"

specifies a unique identifier for the study in which this location is participating. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

### METADATAVERSIONOID="*ODM-identifier*"

specifies the metadata version that is used at the location. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

### EFFECTIVEDATE="*date*"

specifies the date of the metadata version, in compliance with ISO 8601 guidelines. Enclose the date in single or double quotation marks.

**Requirement:** Required.

---

## SIGNATURE Statement

Specifies information about the signatures required in administering the study.

**Restriction:** This is an advanced statement. You must be familiar with CDISC ODM.

**Requirements:** Optional for exporting.

Metadata attributes must be stored in a SAS data set that you reference in the DATA= argument. You cannot specify metadata attributes in the SIGNATURE statement.

---

## Syntax

SIGNATURE DATA=*libref.member-name* ;

**Required Argument****DATA=libref.member-name**

specifies the SAS data set that contains the metadata attributes. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

**SIGNATUREOID="ODM-identifier"**

specifies a unique identifier for the signature. Enclose the identifier in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

**METHODOLOGY=DIGITAL | ELECTRONIC**

specifies the form in which the signature was stored.

**Requirement:** Required.

**MEANING="string"**

specifies information about the context in which the signature has meaning. Enclose the string in single or double quotation marks.

**Requirement:** Required.

**LEGALREASON="string"**

specifies why signature authentication is necessary. Enclose the string in single or double quotation marks. A string of characters up to the maximum OID length that is defined in CDISC ODM is supported.

**Requirement:** Required.

---

**CLINICALDATA Statement**

When importing, identifies the output SAS data set and specifies where the data content in the CDISC ODM XML document begins; when exporting, identifies the input SAS data set and specifies any optional metadata attributes.

**Requirements:** Required for both importing and exporting.  
When exporting, the optional metadata attributes must be specified in the CLINICALDATA statement.

**Syntax****CLINICALDATA**

```
OUT=libref.member-name SASDATASETNAME="name" |
DATA=libref.member-name <metadata-attributes> ;
```

**Summary of Optional Arguments**

```
DATA=libref.member-name
metadata-attributes
OUT=libref.member-name
SASDATASETNAME="name"
```

## Arguments

### OUT=*libref.member-name*

identifies the output SAS data set. A SAS data set is any file that is accessed by SAS, such as a SAS data file or a file that points to data from other sources, such as a DBMS table that is accessed with a SAS/ACCESS engine. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

**Requirement:** Required for importing.

### SASDATASETNAME="*name*"

specifies a CDISC ODM ItemGroupDef attribute that identifies where the data content in the XML document begins. Enclose the name in single or double quotation marks.

**Requirement:** Required for importing.

### DATA=*libref.member-name*

specifies the input SAS data set that contains the clinical data and the required KeySet members to be exported to the XML document. A SAS data set is any file that is accessed by SAS, such as a SAS data file or a file that points to data from other sources, such as a DBMS table that is accessed with a SAS/ACCESS engine. The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

The required KeySet members include the following:

- \_\_STUDYOID
- \_\_METADATAVERSIONOID
- \_\_SUBJECTKEY
- \_\_STUDYEVENTOID
- \_\_STUDYEVENTREPEATKEY
- \_\_FORMOID
- \_\_FORMREPEATKEY
- \_\_ITEMGROUPOID
- \_\_ITEMGROUPREPEATKEY
- \_\_TRANSACTIONTYPE

**Requirement:** Required for exporting.

**See:** [“CDISC ODM KeySet Members” on page 6](#)

### *metadata-attributes*

includes the following syntax, which must be specified in the CLINICALDATA statement:

NAME="*string*"

specifies the study name. Enclose the string in single or double quotation marks.

**Requirement:** Optional for exporting.

DOMAIN="*domain-name*"

ORIGIN="*string*"

PURPOSE="*string*"

COMMENT="*string*"

specifies submission information as defined in CDISC Submission Metadata Model. Enclose each value in single or double quotation marks.

**Requirement:** Optional for exporting.

INVESTIGATORREF=NO | YES

determines whether PROC CDISC imports the unique identifier for the investigator user. INVESTIGATORREF=YES creates the SAS variable `__USEROID` in the imported SAS data set.

**Default:** NO

**Requirements:**

Optional for importing.

To specify INVESTIGATORREF=YES, you must also specify ODMMINIMUMKEYSET=NO in the ODM statement.

SITEREF=NO | YES

determines whether PROC CDISC imports the unique identifier for the study location. SITEREF=YES creates the SAS variable `__LOCATIONOID` in the imported SAS data set.

**Default:** NO

**Requirements:**

Optional for importing.

To specify SITEREF=YES, you must also specify ODMMINIMUMKEYSET=NO in the ODM statement.

---

## CONTENTS Statement

Describes the contents of a CDISC ODM SAS data set in the SAS log.

**Interactions:** To control the line size for the SAS log, use the `LINESIZE=` SAS system option. Use the ODM statement `LONGNAMES=` processing option to determine the sources of captured SAS name parameters and to control the maximum length of SAS name parameters that is valid. See [ODM Statement on page 63](#).

**See:** [“Describing a CDISC ODM SAS Data Set with the CONTENTS Statement” on page 45](#)

---

## Syntax

`CONTENTS TABLE="name";`

### *Required Argument*

`TABLE="name"`

specifies a CDISC ODM SAS data set name, which is an ItemGroupDef attribute that identifies where the data content in the XML document begins. Enclose the name in single or double quotation marks. The source of the captured SAS name is determined by the ODM statement `LONGNAMES=` processing option.

**Requirement:** Required.

---

## DATASETS Statement

Produces a directory listing of the CDISC ODM XML document in the SAS log.

**Interactions:** To control the line size for the SAS log, use the `LINESIZE=` SAS system option.

Use the ODM statement LONGNAMES= processing option to determine the sources of captured SAS name parameters and to control the maximum length of SAS names that is valid. See [ODM Statement on page 63](#).

**See:** [“Listing a Directory with the DATASETS Statement” on page 50](#)

---

## **Syntax**

**DATASETS;**





## Chapter 7

# CDISC SDTM Procedure

---

<b>Overview: CDISC SDTM Procedure</b> .....	<b>83</b>
<b>Syntax: CDISC SDTM Procedure</b> .....	<b>83</b>
PROC CDISC for SDTM Statement .....	84
SDTM Statement .....	84
DOMAINDATA Statement .....	84

---

## Overview: CDISC SDTM Procedure

PROC CDISC performs data content validation on a SAS data set that conforms to CDISC SDTM version 3.1. PROC CDISC validates the SAS data set against domain definitions that are provided by CDISC SDTM.

The following table provides a quick reference for the procedure statements:

**Table 7.1** *Table of Procedure Tasks*

Statement	Task	Required
<a href="#">PROC CDISC for SDTM Statement on page 84</a>	Specifies CDISC SDTM as the model and validates a SAS data set that conforms to CDISC SDTM	yes
<a href="#">SDTM Statement on page 84</a>	Specifies the CDISC SDTM version number	yes
<a href="#">DOMAINDATA Statement on page 84</a>	Identifies the SAS data set to be validated	yes

---

## Syntax: CDISC SDTM Procedure

**Restrictions:** PROC CDISC is supported in the following operating environments: Windows, UNIX, and z/OS.

PROC CDISC supports only one CDISC model and one DOMAINDATA statement per invocation.

---

```
PROC CDISC MODEL=SDTM ;
    SDTM SDTMVersion="version-number" ;
    DOMAINDATA DATA=libref.member-name DOMAIN=domain CATEGORY=category ;
```

---

## PROC CDISC for SDTM Statement

Specifies CDISC SDTM as the model and validates a SAS data set that conforms to CDISC SDTM.

**Requirement:** Required.

---

### Syntax

```
PROC CDISC MODEL=SDTM;
```

### Required Argument

#### MODEL=SDTM

specifies CDISC SDTM as the model.

**Requirement:** Required.

**Interaction:** The model specification determines the syntax for the PROC CDISC statement and subsequent procedure statements. Specifying MODEL=SDTM requires that the next statement must be the SDTM statement.

---

## SDTM Statement

Specifies the CDISC SDTM version number.

**Requirement:** Required.

---

### Syntax

```
SDTM SDTMVersion="version-number";
```

### Required Argument

#### SDTMVersion="*version-number*"

specifies the CDISC SDTM version number. The valid value is 3.1. Enclose the number in single or double quotation marks.

**Requirement:** Required.

---

## DOMAINDATA Statement

Identifies the SAS data set to be validated.

**Requirement:** Required.

## Syntax

**DOMAINDATA** **DATA**=*libref.member-name* **DOMAIN**=*domain* **CATEGORY**=*category*;

### Required Arguments

#### **DATA**=*libref.member-name*

specifies the SAS data set that conforms to CDISC SDTM. A SAS data set is any file that is accessed by SAS, such as a SAS data file or a file that points to data from other sources, such as a DBMS table that is accessed with a SAS/ACCESS engine.

The *libref*, which is assigned with the LIBNAME statement, is an alias for the data storage location where *member-name* is stored.

**Requirement:** Required.

#### **DOMAIN**=*domain-code*

specifies a unique two-character domain code. Each domain is a collection of observations that are common to a specific subject. PROC CDISC currently supports 15 of the domains in CDISC SDTM version 3.1. The trial design components category include the majority of unsupported domains.

The following table lists the codes for the supported domains:

**Table 7.2** CDISC SDTM Domains Supported by PROC CDISC

Supported CDISC SDTM Domain	Code
Demography	DM
Comments	CO
Concomitant Medications	CM
Exposure	EX
Substance Use	SU
Adverse Events	AE
Disposition	DS
Medical History	MH
ECG Test Results	EG
Inclusion/Exclusion Exception	IE
Laboratory Test Results	LB
Physical Examinations	PE
Questionnaires	QS
Subject Characteristics	SC

Supported CDISC SDTM Domain	Code
Vital Signs	VS

**Requirement:** Required.

**CATEGORY=*model-type***

specifies a domain model type. Each model type represents a category from which a domain is derived.

The following table lists the supported model types:

Domain	Model Type
Demography	Special
Comments	Special
Concomitant Medications	Interventions
Exposure	Interventions
Substance Use	Interventions
Adverse Events	Events
Disposition	Events
Medical History	Events
ECG Test Results	Findings
Inclusion/Exclusion Exception	Findings
Laboratory Test Results	Findings
Physical Examinations	Findings
Questionnaires	Findings
Subject Characteristics	Findings
Vital Signs	Findings

**Requirement:** Required.

## Part 4

---

# Appendixes

<i>Appendix 1</i>	
<b>Sample CDISC ODM XML Document</b> .....	89



*Appendix 1*

# Sample CDISC ODM XML Document

This is an example of an XML document that conforms to CDISC ODM.

```

?xml version="1.0" encoding="windows-1252" ?>
<!--
  Clinical Data Interchange Standards Consortium (CDISC)
  Operational Data Model (ODM) for clinical data interchange

  You can learn more about CDISC standards efforts at
  http://www.cdisc.org/standards/index.html
-->

<ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
      xmlns:ds="http://www.w3.org/2000/09/xmlsig#"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 ODM1-2-0.xsd"

      ODMVersion="1.2"
      FileOID="000-00-0000"
      FileType="Snapshot"
      Description="Adverse events from the CTChicago file"

      AsOfDateTime="2009-03-31T14:01:41"
      CreationDateTime="2009-03-31T14:01:41">

  <Study OID="STUDY.StudyOID">

    <!--
      GlobalVariables is a REQUIRED section in ODM markup
    -->
    <GlobalVariables>
      <StudyName>CDISC Connect-A-Thon Test Study III</StudyName>
      <StudyDescription>This file contains test data from a previous CDISC Connect-A-Thon.</
StudyDescription>
      <ProtocolName>CDISC-Protocol-00-000</ProtocolName>
    </GlobalVariables>

    <BasicDefinitions />

    <!--
      Internal ODM markup required metadata
    -->
    <MetaDataVersion OID="v1.1.0" Name="Version 1.1.0">
      <Protocol>
        <StudyEventRef StudyEventOID="SE.VISIT1" OrderNumber="1" Mandatory="Yes" />
      </Protocol>

      <StudyEventDef OID="SE.VISIT1" Name="Study Event Definition" Repeating="Yes" Type="Common">
        <FormRef FormOID="FORM.AE" OrderNumber="1" Mandatory="No" />
      </StudyEventDef>

      <FormDef OID="FORM.AE" Name="Form Definition" Repeating="Yes">
        <ItemGroupRef ItemGroupOID="IG.AE" Mandatory="No" />
      </FormDef>

    <!--
      Columns defined in the table
    -->
    <ItemGroupDef OID="IG.AE" Repeating="Yes"
      SASDatasetName="AE"
      Name="Adverse Events"
      Domain="AE"
      Comment="Some adverse events from this trial">

```



```

    <ItemRef ItemOID="ID.TAREA"      OrderNumber="1"  Mandatory="No" />
    <ItemRef ItemOID="ID.PNO"        OrderNumber="2"  Mandatory="No" />
    <ItemRef ItemOID="ID.SCTRY"      OrderNumber="3"  Mandatory="No" />
    <ItemRef ItemOID="ID.F_STATUS"   OrderNumber="4"  Mandatory="No" />
    <ItemRef ItemOID="ID.LINE_NO"    OrderNumber="5"  Mandatory="No" />
    <ItemRef ItemOID="ID.AETERM"     OrderNumber="6"  Mandatory="No" />
    <ItemRef ItemOID="ID.AESTMON"    OrderNumber="7"  Mandatory="No" />
    <ItemRef ItemOID="ID.AESTDAY"    OrderNumber="8"  Mandatory="No" />
    <ItemRef ItemOID="ID.AESTYR"     OrderNumber="9"  Mandatory="No" />
    <ItemRef ItemOID="ID.AESTDT"    OrderNumber="10" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENMON"    OrderNumber="11" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENDAY"    OrderNumber="12" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENYR"     OrderNumber="13" Mandatory="No" />
    <ItemRef ItemOID="ID.AEENDT"     OrderNumber="14" Mandatory="No" />
    <ItemRef ItemOID="ID.AESEV"      OrderNumber="15" Mandatory="No" />
    <ItemRef ItemOID="ID.AEREL"      OrderNumber="16" Mandatory="No" />
    <ItemRef ItemOID="ID.AEOUT"      OrderNumber="17" Mandatory="No" />
    <ItemRef ItemOID="ID.AEACTTRT"   OrderNumber="18" Mandatory="No" />
    <ItemRef ItemOID="ID.AECONTRT"   OrderNumber="19" Mandatory="No" />
  </ItemGroupDef>

  <!--
    Column attributes as defined in the table
  -->
  <AreaDef OID="ID.TAREA"      SASFieldName="TAREA"      Name="Therapeutic
Area"      DataType="text"      Length="4">
    <CodeListRef CodeListOID="CL.$TAREAF" /
  >
  </
ItemDef>

    <ItemDef OID="ID.PNO"      SASFieldName="PNO"      Name="Protocol
Number"    DataType="text"      Length="15" />
    <ItemDef OID="ID.SCTRY"    SASFieldName="SCTRY"
Name="Country"      DataType="text"      Length="4">
    <CodeListRef CodeListOID="CL.$SCTRYF" /
  >
  </
ItemDef>

    <ItemDef OID="ID.F_STATUS" SASFieldName="F_STATUS" Name="Record status, 5 levels, internal
use" DataType="text"      Length="1">
    <CodeListRef CodeListOID="CL.$F_STATU" />
  </ItemDef>
    <ItemDef OID="ID.LINE_NO"  SASFieldName="LINE_NO"  Name="Line
Number"    DataType="integer" Length="2" />
    <ItemDef OID="ID.AETERM"   SASFieldName="AETERM"  Name="Conmed
Indication"      DataType="text"      Length="100" />
    <ItemDef OID="ID.AESTMON"  SASFieldName="AESTMON"  Name="Start Month - Enter Two Digits
01-12"  DataType="integer" Length="2" />
    <ItemDef OID="ID.AESTDAY"  SASFieldName="AESTDAY"  Name="Start Day - Enter Two Digits
01-31"  DataType="integer" Length="2" />
    <ItemDef OID="ID.AESTYR"   SASFieldName="AESTYR"   Name="Start Year - Enter Four Digit
Year"      DataType="integer" Length="4" />
    <ItemDef OID="ID.AESTDT"   SASFieldName="AESTDT"   Name="Derived Start
Date"      DataType="date" />
    <ItemDef OID="ID.AEENMON"  SASFieldName="AEENMON"  Name="Stop Month - Enter Two Digits
01-12"  DataType="integer" Length="2" />
    <ItemDef OID="ID.AEENDAY"  SASFieldName="AEENDAY"  Name="Stop Day - Enter Two Digits
01-31"  DataType="integer" Length="2" />
    <ItemDef OID="ID.AEENYR"   SASFieldName="AEENYR"   Name="Stop Year - Enter Four Digit
Year"      DataType="integer" Length="4" />
    <ItemDef OID="ID.AEENDT"   SASFieldName="AEENDT"   Name="Derived Stop
Date"      DataType="date" />
    <ItemDef OID="ID.AESEV"    SASFieldName="AESEV"
Name="Severity"      DataType="text"      Length="1">
    <CodeListRef CodeListOID="CL.$AESEV" />
  </ItemDef>

```

```

      <ItemDef OID="ID.AEREL" SASFieldName="AEREL" Name="Relationship to study
drug"      DataType="text" Length="1">
      <CodeListRef CodeListOID="CL.$AEREL" /
>
    </
ItemDef>

      <ItemDef OID="ID.AEOUT" SASFieldName="AEOUT"
Name="Outcome"      DataType="text" Length="1">
      <CodeListRef CodeListOID="CL.$AEOUT" /
>
    </
ItemDef>

      <ItemDef OID="ID.AEACTTRT" SASFieldName="AEACTTRT" Name="Actions taken re study
drug"      DataType="text" Length="1">
      <CodeListRef CodeListOID="CL.$AEACTTR" /
>
    </
ItemDef>

      <ItemDef OID="ID.AECONTRT" SASFieldName="AECONTRT" Name="Actions taken,
other"      DataType="text" Length="1">
      <CodeListRef CodeListOID="CL.$AECONTR" />
    </ItemDef>

    <!--
      Translation to ODM markup for any PROC FORMAT style
      user defined or SAS internal formatting specifications
      applied to columns in the table
    -->
    <CodeList OID="CL.$TAREAF" SASFormatName="$TAREAF" Name="$TAREAF" DataType="text">
      <CodeListItem CodedValue='ONC'>
        <Decode>
          <TranslatedText xml:lang="en">Oncology</TranslatedText>
        </Decode>
      </CodeListItem>
    </CodeList>

    <CodeList OID="CL.$SCTRYF" SASFormatName="$SCTRYF" Name="$SCTRYF" DataType="text">
      <CodeListItem CodedValue='USA'>
        <Decode>
          <TranslatedText xml:lang="en">United States</TranslatedText>
        </Decode>
      </CodeListItem>
    </CodeList>

    <CodeList OID="CL.$F_STATU" SASFormatName="$F_STATU" Name="$F_STATU" DataType="text">
      <CodeListItem CodedValue='S'>
        <Decode>
          <TranslatedText xml:lang="en">Source verified, not queried</TranslatedText>
        </Decode>
      </CodeListItem>
      <CodeListItem CodedValue='V'>
        <Decode>
          <TranslatedText xml:lang="en">Source verified, queried</TranslatedText>
        </Decode>
      </CodeListItem>
    </CodeList>

```

```

<CodeList OID="CL.$AESEV" SASFormatName="$AESEV" Name="$AESEV" DataType="text">
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText xml:lang="en">Mild</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText xml:lang="en">Moderate</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText xml:lang="en">Severe</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='4'>
    <Decode>
      <TranslatedText xml:lang="en">Life Threatening</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

<CodeList OID="CL.$AEREL" SASFormatName="$AEREL" Name="$AEREL" DataType="text">
  <CodeListItem CodedValue='0'>
    <Decode>
      <TranslatedText xml:lang="en">None</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText xml:lang="en">Unlikely</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText xml:lang="en">Possible</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText xml:lang="en">Probable</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

<CodeList OID="CL.$AEOUT" SASFormatName="$AEOUT" Name="$AEOUT" DataType="text">
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText xml:lang="en">Resolved, no residual effects</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText xml:lang="en">Continuing</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText xml:lang="en">Resolved, residual effects</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='4'>
    <Decode>
      <TranslatedText xml:lang="en">Death</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

```

```

<CodeList OID="CL.$AEACTTR" SASFormatName="$AEACTTR" Name="$AEACTTR" DataType="text">
  <CodeListItem CodedValue='0'>
    <Decode>
      <TranslatedText xml:lang="en">None</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText xml:lang="en">Discontinued permanently</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText xml:lang="en">Reduced</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText xml:lang="en">Interrupted</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>

<CodeList OID="CL.$AECONTR" SASFormatName="$AECONTR" Name="$AECONTR" DataType="text">
  <CodeListItem CodedValue='0'>
    <Decode>
      <TranslatedText xml:lang="en">None</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText xml:lang="en">Medication required</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText xml:lang="en">Hospitalization required or prolonged</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText xml:lang="en">Other</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>
</MetaDataVersion>
</Study>

```

```

<!--
  Administrative metadata
-->
<AdminData />

<!--
  Clinical Data   : AE
                  Adverse Events
                  Some adverse events from this trial
-->
<ClinicalData StudyOID="STUDY.StudyOID" MetaDataVersionOID="v1.1.0">
  <SubjectData SubjectKey="001">
    <StudyEventData StudyEventOID="SE.VISIT1" StudyEventRepeatKey="1">
      <FormData FormOID="FORM.AE" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="1">
          <ItemData ItemOID="ID.TAREA" Value="ONC" />
          <ItemData ItemOID="ID.PNO" Value="143-02" />
          <ItemData ItemOID="ID.SCTRY" Value="USA" />
          <ItemData ItemOID="ID.F_STATUS" Value="V" />
          <ItemData ItemOID="ID.LINE_NO" Value="1" />
          <ItemData ItemOID="ID.AETERM" Value="HEADACHE" />
          <ItemData ItemOID="ID.AESTMON" Value="06" />
          <ItemData ItemOID="ID.AESTDAY" Value="10" />
          <ItemData ItemOID="ID.AESTYR" Value="1999" />
          <ItemData ItemOID="ID.AESTDT" Value="1999-06-10" />
          <ItemData ItemOID="ID.AEENMON" Value="06" />
          <ItemData ItemOID="ID.AEENDAY" Value="14" />
          <ItemData ItemOID="ID.AEENYR" Value="1999" />
          <ItemData ItemOID="ID.AEENDT" Value="1999-06-14" />
          <ItemData ItemOID="ID.AESEV" Value="1" />
          <ItemData ItemOID="ID.AEREL" Value="0" />
          <ItemData ItemOID="ID.AEOUT" Value="1" />
          <ItemData ItemOID="ID.AEACTTRT" Value="0" />
          <ItemData ItemOID="ID.AECONTRT" Value="1" />
        </ItemGroupData>
        <ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="2">
          <ItemData ItemOID="ID.TAREA" Value="ONC" />
          <ItemData ItemOID="ID.PNO" Value="143-02" />
          <ItemData ItemOID="ID.SCTRY" Value="USA" />
          <ItemData ItemOID="ID.F_STATUS" Value="V" />
          <ItemData ItemOID="ID.LINE_NO" Value="2" />
          <ItemData ItemOID="ID.AETERM" Value="CONGESTION" />
          <ItemData ItemOID="ID.AESTMON" Value="06" />
          <ItemData ItemOID="ID.AESTDAY" Value="11" />
          <ItemData ItemOID="ID.AESTYR" Value="1999" />
          <ItemData ItemOID="ID.AESTDT" Value="1999-06-11" />
          <ItemData ItemOID="ID.AEENMON" Value="" />
          <ItemData ItemOID="ID.AEENDAY" Value="" />
          <ItemData ItemOID="ID.AEENYR" Value="" />
          <ItemData ItemOID="ID.AEENDT" Value="" />
          <ItemData ItemOID="ID.AESEV" Value="1" />
          <ItemData ItemOID="ID.AEREL" Value="0" />
          <ItemData ItemOID="ID.AEOUT" Value="2" />
          <ItemData ItemOID="ID.AEACTTRT" Value="0" />
          <ItemData ItemOID="ID.AECONTRT" Value="1" />
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>

```



# Index

---

## A

accessibility features [4](#)

## B

BASICDEFINITIONS statement [71](#)

## C

CDISC models [3](#)

CDISC ODM [5](#)

    KeySet members [6](#)

    metadata attributes [9](#)

    PROC CDISC statement [62](#)

    syntax [60](#)

CDISC ODM examples

    describing data sets with CONTENTS  
    statement [45](#)

    exporting with metadata attributes in  
    data sets [37](#)

    exporting with metadata attributes in  
    statement syntax [29](#)

    exporting with optional statements [42](#)

    importing with default KeySet

        processing [18](#)

    importing with KeySet processing

        options [20](#)

    importing with language identifiers [22](#)

    importing with OrderNumber attributes  
    [28](#)

    listing a directory with DATASETS  
    statement [50](#)

CDISC overview [3](#)

CDISC procedure [3](#)

    CDISC ODM [60](#)

    CDISC SDTM syntax [83](#)

    syntax overview [4](#)

CDISC SDTM [13](#)

    PROC CDISC statement [84](#)

    syntax [83](#)

    validating data sets [13](#)

CDISC SDTM examples

    validating data sets [53](#)

    validating Oracle tables [54](#)

CLINICALDATA statement [78](#)

CONTENTS statement [45, 80](#)

## D

data sets

    describing with CONTENTS statement  
    [45](#)

    validating for CDISC SDTM [53](#)

DATASETS statement [50, 80](#)

directory listing [50](#)

DOMAINDATA statement [84](#)

## E

exporting

    CDISC ODM, with metadata attributes  
    in data sets [37](#)

    CDISC ODM, with metadata attributes  
    in statement syntax [29](#)

    CDISC ODM, with optional statements  
    [42](#)

## G

GLOBALVARIABLES statement [71](#)

## I

importing

    CDISC ODM, with default KeySet  
    processing [18](#)

    CDISC ODM, with KeySet processing  
    options [20](#)

    CDISC ODM, with language identifiers  
    [22](#)

    CDISC ODM, with OrderNumber  
    attributes [28](#)

**K**

KeySet members [6](#), [18](#), [20](#)

**L**

language identifiers [22](#)

LOCATION statement [76](#)

**M**

metadata attributes [29](#), [37](#)

metadata attributes for CDISC ODM [9](#)

METADATAVERSION statement [72](#)

**O**

ODM statement [63](#)

Oracle tables

    validating for CDISC SDTM [54](#)

OrderNumber attributes [28](#)

**P**

PRESENTATION statement [73](#)

PROC CDISC for ODM statement [61](#)

PROC CDISC for SDTM statement [84](#)

PROC CDISC statement

    CDISC ODM [62](#)

    CDISC SDTM [84](#)

**S**

SDTM statement [84](#)

SIGNATURE statement [77](#)

statements (optional) [42](#)

STUDY statement [70](#)

**U**

USER statement [74](#)

**V**

validating

    CDISC SDTM data sets [13](#), [53](#)

    CDISC SDTM Oracle tables [54](#)

**X**

XML documents [5](#)