# Contents

## PART 1 Concepts  

**Chapter 1** • Introduction to the CDISC Procedure  
- What Is CDISC?  
- What Is the CDISC Procedure?  
- Overview of PROC CDISC Syntax  
- Accessibility Features of the CDISC Procedure

## PART 2 Usage  

**Chapter 2** • Importing and Exporting a CDISC ODM XML Document  
- Introduction to CDISC ODM  
- CDISC ODM KeySet Members  
- Specifying CDISC ODM Metadata Attributes

**Chapter 3** • Validating a CDISC SDTM SAS Data Set  
- Introduction to CDISC SDTM  
- Validating a CDISC SDTM SAS Data Set  
- CDISC Procedure Capabilities for Validating a CDISC SDTM SAS Data Set

## PART 3 Procedure Reference  

**Chapter 4** • CDISC ODM Procedure  
- Overview: CDISC ODM Procedure  
- Syntax: CDISC ODM Procedure
Chapter 7 • CDISC SDTM Procedure ....................................................... 85
Overview: CDISC SDTM Procedure .................................................... 85
Syntax: CDISC SDTM Procedure ......................................................... 85

PART 4 Appendixes  91

Appendix 1 • Sample CDISC ODM XML Document ........................ 93

Recommended Reading ................................................................. 101
Index ......................................................................................... 103
Part 1

Concepts

Chapter 1
   Introduction to the CDISC Procedure .................................. 3

Chapter 2
   Importing and Exporting a CDISC ODM XML Document ............. 5

Chapter 3
   Validating a CDISC SDTM SAS Data Set .............................. 13
Chapter 1
Introduction to the CDISC Procedure

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 website at www.cdisc.org.

Note: The CDISC mission statement above is cited from the CDISC website. 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 CDISC 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.

**T I P** 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 of these statements 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:

```sas
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:

```sas
proc cdisc model=sdtm;
   sdtm sdtmversion="3.1";
   domaindata data=Results.AE domain=AE category=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. However, the product might 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 or call SAS Technical Support.
Chapter 2
Importing and Exporting a CDISC ODM XML Document

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 and metadata 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:

```xml
<ClinicalData StudyOID="123-456-789" MetaDataVersionOID="v1.1.0">
```
### 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**

<table>
<thead>
<tr>
<th>KeySet Member</th>
<th>Description</th>
<th>Resulting SAS Variable Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudyOID=</td>
<td>Specifies a unique identifier for the study.</td>
<td>_STUDYOID</td>
</tr>
<tr>
<td>MetaDataVersionOID=</td>
<td>Specifies the metadata version that is used by the study.</td>
<td>_METADATAVERSIONOID</td>
</tr>
<tr>
<td>SubjectKey=</td>
<td>Specifies a subject within the study.</td>
<td>_SUBJECTKEY</td>
</tr>
<tr>
<td>StudyEventOID=</td>
<td>Specifies a StudyEventDef within the study.</td>
<td>_STUDYEVENTOID</td>
</tr>
<tr>
<td>StudyEventRepeatKey=</td>
<td>Specifies a study event repeat key.</td>
<td>_STUDYEVENTREPEATKEY</td>
</tr>
<tr>
<td>FormOID=</td>
<td>Specifies a form used in the study.</td>
<td>_FORMOID</td>
</tr>
<tr>
<td>FormRepeatKey=</td>
<td>Distinguishes between repeats of the same type of form in a single study event.</td>
<td>_FORMREPEATKEY</td>
</tr>
<tr>
<td>ItemGroupOID=</td>
<td>Specifies an ItemGroup in the study.</td>
<td>_ITEMGROUPOID</td>
</tr>
<tr>
<td>ItemGroupRepeatKey=</td>
<td>Specifies an ItemGroup repeat key.</td>
<td>_ITEMGROUPREPEATKEY</td>
</tr>
</tbody>
</table>
### Table 2.2  Exporting KeySet Members from a SAS Data Set

<table>
<thead>
<tr>
<th>KeySet Member</th>
<th>Description</th>
<th>Resulting KeySet Attributes</th>
<th>Created By</th>
</tr>
</thead>
<tbody>
<tr>
<td>__STUDYOID</td>
<td>Specifies a unique identifier for the study.</td>
<td>StudyOID=</td>
<td>STUDY statement</td>
</tr>
<tr>
<td>__METADATAVEVISIONOID</td>
<td>Specifies the metadata version that is used by the study.</td>
<td>MetaDataVersion OID=</td>
<td>METADATAVERSION statement</td>
</tr>
<tr>
<td>__SUBJECTKEY</td>
<td>Specifies a subject within the study.</td>
<td>SubjectKey=</td>
<td>Read from the exported SAS data set</td>
</tr>
<tr>
<td>__STUDYEVENTOID</td>
<td>Specifies a StudyEventDef within the study.</td>
<td>StudyEventOID=</td>
<td>STUDY statement</td>
</tr>
<tr>
<td>__STUDYEVENTREPEATKEY</td>
<td>Specifies a study event repeat key.</td>
<td>StudyEventRepeat Key=</td>
<td>Automatically generated</td>
</tr>
<tr>
<td>__FORMOID</td>
<td>Specifies a form used in the study.</td>
<td>FormOID=</td>
<td>Automatically generated</td>
</tr>
<tr>
<td>__FORMREPEATKEY</td>
<td>Distinguishes between repeats of the same type of form in a single study event.</td>
<td>FormRepeatKey=</td>
<td>Automatically generated</td>
</tr>
<tr>
<td>__ITEMGROUPOID</td>
<td>Specifies an ItemGroup in the study.</td>
<td>ItemGroupOID=</td>
<td>Automatically generated</td>
</tr>
<tr>
<td>__ITEMGROUPREPEATKEY</td>
<td>Specifies an ItemGroup repeat key.</td>
<td>ItemGroupRepeat Key=</td>
<td>Automatically generated</td>
</tr>
</tbody>
</table>
### 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 that is valid for SAS names.

**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 64.

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:

```plaintext
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:

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

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

```sas
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:

```sas
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 attributes for the version 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 attributes for the ODM statement is specified in the statement.

```sas
filename Xmlinp 'CDISC-ODM-XML-document';
proc cdisc model=odm read=Xmlinp formatactive=yes formatnoreplace=no;
odm odmversion="1.2" odmmimumoidlength=16 odmmimumkeyset=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 ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION statements 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.

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

- Metadata attributes for the optional BASICDEFINITIONS, PRESENTATION, USER, LOCATION, and SIGNATURE statements must be stored in a SAS data set that you reference in the DATA= argument.
Chapter 3
Validating a CDISC SDTM SAS Data Set

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 Submission 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. 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.
- DOMAININDATA 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 STDM 3.1 data content validation only.
Part 2

Usage

Chapter 4
CDISC Procedure Examples for CDISC ODM ........................ 17

Chapter 5
CDISC Procedure Examples for CDISC SDTM ....................... 53
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. It also 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 Appendix 1, “Sample CDISC ODM XML Document,” on page 93.

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 file reference 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.
   • File reference 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 names 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 (which is AE) begins.

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
importing a cdisc odm xml document using default keyset processing

```
proc cdisc model=odm
   read=Xmlinp
   formatactive=yes
   formatnoreplace=no;
   odm odmversion="1.2" odmmimumkeyset=no;
   clinicaldata out=Results.AE sasdatasetname="AE";
run;

proc contents data=Results.AE varnum;
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 the resulting SAS variable names are two underscores.)

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

### Output 4.1: PROC CONTENTS Output for Results.AE

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
<th>Informat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>__StudyOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>__MetaDataVersionOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>__SubjectKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>__StudyEventOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>__StudyEventRepeatKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>__FormOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>__FormRepeatKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>__ItemGroupOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>__ItemGroupRepeatKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>__TransactionType</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>TAREA</td>
<td>Char</td>
<td>4</td>
<td>$TAREA.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>FNO</td>
<td>Char</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>SCTRY</td>
<td>Char</td>
<td>4</td>
<td>$SCTRY.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>F_STATUS</td>
<td>Char</td>
<td>1</td>
<td>$F_STATU</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>LINE_NO</td>
<td>Num</td>
<td>8</td>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>AETERM</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>AESMON</td>
<td>Num</td>
<td>8</td>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>AESDAY</td>
<td>Num</td>
<td>8</td>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>AESTYR</td>
<td>Num</td>
<td>8</td>
<td>4.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>AESTDT</td>
<td>Num</td>
<td>8</td>
<td>DATE.</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>AEENMON</td>
<td>Num</td>
<td>8</td>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>AEENDAY</td>
<td>Num</td>
<td>8</td>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>AEENYR</td>
<td>Num</td>
<td>8</td>
<td>4.</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>AEENDT</td>
<td>Num</td>
<td>8</td>
<td>DATE.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>AEEV</td>
<td>Char</td>
<td>1</td>
<td>$AEEV.</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>AEREL</td>
<td>Char</td>
<td>1</td>
<td>$AEREL</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>AEOUT</td>
<td>Char</td>
<td>1</td>
<td>$AEOUT.</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>AEACTTRT</td>
<td>Char</td>
<td>1</td>
<td>$AEACTTR.</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>AECONTRT</td>
<td>Char</td>
<td>1</td>
<td>$AECONTR.</td>
<td></td>
</tr>
</tbody>
</table>
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 Appendix 1, “Sample CDISC ODM XML Document,” on page 93.

**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 file reference 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.
   - File reference 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 names as the converted formats.

4. **ODDMINIMUMKEYSET=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 (which is AE) begins.

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.

```sas
libname Results 'C:\My Documents\';  
filename Xmlinp 'C:\XML\AE.xml';  
proc cdisc model=odm  
    read=Xmlinp  
    formatactive=yes  
    formatnoreplace=no;  
    odm odmversion="1.2"  
        odmminimumkeyset=yes  
        odmmaximumoidlength=18;  
    clinicaldata out=Results.AEMin sasdatasetname="AE";  
run;
```
filename Xmlinp clear;

proc contents data=Results.AEMin varnum;
   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.

Output 4.2 PROC CONTENTS Output for Results.AEMin

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
<th>Informat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SubjectKey</td>
<td>Char</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TAREA</td>
<td>Char</td>
<td>4</td>
<td>$TAREA.F.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>FNO</td>
<td>Char</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SCTRY</td>
<td>Char</td>
<td>4</td>
<td>$SCTRY.F.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>F_STATUS</td>
<td>Char</td>
<td>1</td>
<td>$F_STATUS.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>LINE_NO</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>2.</td>
</tr>
<tr>
<td>7</td>
<td>AETERM</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>AESTMON</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>2.</td>
</tr>
<tr>
<td>9</td>
<td>AESTDAY</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>2.</td>
</tr>
<tr>
<td>10</td>
<td>AESTYR</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>4.</td>
</tr>
<tr>
<td>11</td>
<td>AESTDT</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>DATE.</td>
</tr>
<tr>
<td>12</td>
<td>AENMON</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>2.</td>
</tr>
<tr>
<td>13</td>
<td>AEENDAY</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>2.</td>
</tr>
<tr>
<td>14</td>
<td>AENNYR</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>4.</td>
</tr>
<tr>
<td>15</td>
<td>AEENDT</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>DATE.</td>
</tr>
<tr>
<td>16</td>
<td>AESEV</td>
<td>Char</td>
<td>1</td>
<td></td>
<td>$AESEV.</td>
</tr>
<tr>
<td>17</td>
<td>AEREL</td>
<td>Char</td>
<td>1</td>
<td></td>
<td>$AEREL.</td>
</tr>
<tr>
<td>18</td>
<td>AEOUT</td>
<td>Char</td>
<td>1</td>
<td></td>
<td>$AEOUT.</td>
</tr>
<tr>
<td>19</td>
<td>AEACTTR</td>
<td>Char</td>
<td>1</td>
<td></td>
<td>$AEACTTR.</td>
</tr>
<tr>
<td>20</td>
<td>AECONTR</td>
<td>Char</td>
<td>1</td>
<td></td>
<td>$AECONTR.</td>
</tr>
</tbody>
</table>
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 element 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.

The following example imports the XML document:

```xml
<?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>

<!--
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">
  <CodeList.Item CodedValue="1">
    <Decode>
      <TranslatedText xml:lang="de">einz</TranslatedText>
      <TranslatedText xml:lang="en">one</TranslatedText>
      <TranslatedText xml:lang="es">uno</TranslatedText>
    </Decode>
  </CodeList.Item>
  <CodeList.Item CodedValue="2">
    <Decode>
      <TranslatedText xml:lang="de">zwei</TranslatedText>
      <TranslatedText xml:lang="en">two</TranslatedText>
      <TranslatedText xml:lang="es">dos</TranslatedText>
    </Decode>
  </CodeList.Item>
</CodeList>
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 file reference 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.
   - File reference 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 names 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. It creates a SAS format by using the TranslatedText element with the matching language tag attribute. The created SAS format is then applied to the data that is imported from the XML document.

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.Numbers) and specifies the CDISC ODM ItemGroupDef attribute that indicates where the data content in the XML document (which is X) begins.

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\';
filename Xmlinp 'C:\XML\Numbers.xml';
proc cdisc model=odm
   read=Xmlinp
   formatactive=yes
   formatnoreplace=no
   language="de";
   odm odmversion="1.2"  odmminimumkeyset=no;
   clinicaldata out=Results.Numbers sasdatasetname="X";
run;
filename Xmlinp clear;
proc contents data=Results.Numbers;
run;
proc print data=Results.Numbers;
   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.
Output 4.3  PROC CONTENTS Output for Results.Numbers

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
<th>Informat</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>X</td>
<td>Num</td>
<td>8</td>
<td>NUMBERS</td>
<td>12.2</td>
<td>x</td>
</tr>
<tr>
<td>6</td>
<td>FormOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>FormRepeatKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ItemGroupOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ItemGroupRepeatKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MetaDataVersionOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>StudyEventOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>StudyEventRepeatKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>StudyOID</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SubjectKey</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TransactionType</td>
<td>Char</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 element with the matching language tag attribute DE. It applies the SAS format NUMBERS to the data that is imported from the XML document (which is 3). The result is the German word drei.

Output 4.4  PROC PRINT Output for Variable X

The SAS System

<table>
<thead>
<tr>
<th>Obs</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>drei</td>
</tr>
</tbody>
</table>

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 a warning 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 file reference 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.
   - File reference 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 (which is AE) begins.

```sas
libname Results 'C:\My Documents\';  
filename Xmlinp 'C:\XML\AE.xml';  
proc cdisc model=odm read=Xmlinp;  
odm odmversion="1.2" ordernumber=yes;  
   clinicaldata out=Results.AE sasdatasetname="AE";  
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.
Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax

Overview

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

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 file reference Xmlinp 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.
   - File reference Xmlinp, 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.
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\';
filename Xmlout 'C:\XML\AEfull.xml';
proc sort data=Results.AE;
  by __subjectkey;
run:
proc cdisc model=odm
  write=Xmlout
  formatactive=yes
  formatnoreplace=no;
  odm odmversion="1.2"
  fileoid="000-00-0000"
  filetype=SNAPSHOT
  description="Adverse events from the CTChicago file"
  study studyoid="STUDY.StudyOID";
  globalvariables studyname="CDISC Connect-A-Thon Test Study III"
  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"
  name="Version 1.1.0"
  clinicaldata data=Results.AE
    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.
<!-- 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 http://www.cdisc.org/ns/odm/v1.2-0.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>

<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>

</Study>

</ODM>
Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax

```xml
<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.AEEREL" OrderNumber="15" Mandatory="No" />
<ItemRef ItemOID="ID.AEOUT" OrderNumber="16" Mandatory="No" />
<ItemRef ItemOID="ID.AEACTTRT" OrderNumber="17" Mandatory="No" />
<ItemRef ItemOID="ID.AECONTRT" OrderNumber="18" 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" />
</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.AEEREL" Name="AEEREL" SASFieldName="AEEREL" 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" />
</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="$AESEV" DataType="text">
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText>Mild</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>
    <Decode>
      <TranslatedText>Moderate</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='3'>
    <Decode>
      <TranslatedText>Severe</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='4'>
    <Decode>
      <TranslatedText>Life Threatening</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>
<CodeList OID="CL.$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>

<ClinicalData StudyOID="STUDY.Study OID" 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.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>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets

Overview

This example specifies the metadata attributes that are stored in SAS data sets and referenced in the DATA= argument. This example includes only the required PROC CDISC statements for exporting. 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";
```

Then, reference the SAS data sets in PROC CDISC statements:

1. The FILENAME statement assigns the file reference Xmlout to the physical location of the output XML document.
2. The PROC CDISC statement specifies the following:
   - CDISC ODM as the model.
   - File reference Xmlout, which references the physical location of the output XML document to be exported.
3. The 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 STUDY statement includes the DATA= argument to reference the SAS data set Current.Study, which stores the study identifier.
5. The GLOBALVARIABLES statement includes the DATA= argument to reference the SAS data set Current.Globals, which stores general summary information about the study.
6. The METADATAVERSION statement includes the DATA= argument to reference the SAS data set Current.Metadata, which stores the metadata version and version name.
7. The 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.

```sas
filename Xmlout 'C:\XML\AEds.xml';
proc cdisc model=odm write=Xmlout;
   odm data=Current.Odm;
   study data=Current.Study;
   globalvariables data=Current.Globals;
   metadataversion data=Current.Metadata;
   clinicaldata data=Current.AE
      domain="AE"
      name="Adverse Events"
```

Chapter 4 • CDISC Procedure Examples for CDISC ODM
Output

The following output shows the exported XML document:
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

<?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="9.02.02MODD1222009 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>  
</Study OID="STUDY.StudyOID">  
<!--  
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" />  
</ItemGroupRef>  
</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.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" Value="0" />
<ItemData ItemOID="ID.AEENDAY" Value="0" />
<ItemData ItemOID="ID.AEENYR" Value="0" />
<ItemData ItemOID="ID.AEENDT" Value="0" />
<ItemData ItemOID="ID.AESEV" Value="0" />
<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>
This example includes required and optional PROC CDISC statements for exporting a CDISC ODM XML document. The example specifies 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 METADATA VERSION statements either directly in the CDISC procedure statement or stored in a SAS data set.
- You must submit metadata attributes for the optional BASICDEFINITIONS, PRESENTATION, USER, LOCATION, and SIGNATURE statements in SAS data sets referenced in the DATA= argument.
- You must specify optional metadata attributes for the required CLINICALDATA statement as part of the CLINICALDATA statement syntax.

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";
```
data Current.Present;
length TranslatedText $40.;
  PresentationOID="PRES.EN";
  Lang="en";
  TranslatedText="English: Presentation";
output;
  PresentationOID="PRES.SP";
  Lang="sp";
  TranslatedText="Spanish: Presentación";
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";
Then, reference the SAS data sets in PROC CDISC statements:

1. The FILENAME statement assigns the file reference Xmlout to the physical location of the output XML document.

2. The PROC CDISC statement specifies the following:
   - CDISC ODM as the model.
   - File reference Xmlout, which references the physical location of the output XML document to be exported.

3. The 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 STUDY statement includes the DATA= argument to reference the SAS data set Current.Study, which stores the study identifier.

5. The GLOBALVARIABLES statement includes the DATA= argument to reference the SAS data set Current.Globals, which stores general summary information about the study.

6. The 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 METADATAVERSION statement includes the DATA= argument to reference the SAS data set Current.Metadata, which stores the metadata version and version name that are used by the study.

8. The 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 USER statement includes the DATA= argument to reference the SAS data set Current.User, which stores information about users who are involved in the study.
10. The 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 SIGNATURE statement includes the DATA= argument to reference the SAS data set Current.Signature, which stores information about the signatures that are required for administering the study.

12. The 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.

```sas
filename Xmlout 'C:\XML\AEopts.xml';
proc cdisc model=odm write=Xmlout;
   odm data=Current.Odm;
   study data=Current.Study;
   globalvariables data=Current.Globals;
   basicdefinitions data=Current.Basic;
   metadataversion data=Current.Metadata;
   presentation data=Current.Present;
   user data=Current.User;
   location data=Current.Location;
   signature data=Current.Signature;
   clinicaldata data=Current.AE
      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 lists the attributes of a CDISC ODM SAS data set in the CDISC ODM XML document named AE.XML. The output is displayed in the SAS log.

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

To view the AE.XML document, see Appendix 1, “Sample CDISC ODM XML Document,” on page 93.

**Program with LONGNAMES=NO**

The following SAS program lists the attributes of a CDISC ODM SAS data set using the ODM statement LONGNAMES=NO processing option:
1. The FILENAME statement assigns the file reference Xmlinp to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.

2. The PROC CDISC statement specifies the following:
   - CDISC ODM as the model.
   - File reference 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. Names 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= or SDSVarName= attribute in ODM ItemDef elements. SAS format names are captured from the SASFormatName= attribute in the ODM CodeList elements.

4. The CONTENTS statement writes the contents of the CDISC ODM SAS data set in the SAS log. The CONTENTS statement must specify the name that is in the SASDatasetName= attribute in the ODM ItemGroupDef element.

```sas
filename Xmlinp 'C:\XML\AE.xml';
proc cdisc model=odm read=Xmlinp;
   odm odmversion="1.2" longnames=no;
   contents table="AE";
run;
```
Log 4.2  Contents of CDISC ODM SAS Data Set with LONGNAMES=NO

Contents of CDISC ODM Table

<table>
<thead>
<tr>
<th>Name</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Libref</td>
<td>XMLINP</td>
</tr>
<tr>
<td>ODM Version</td>
<td>1.2</td>
</tr>
<tr>
<td>AsOfDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>CreationDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>SourceSystem</td>
<td>SAS 9.1</td>
</tr>
<tr>
<td>SourceSystemVersion</td>
<td>9.01.01M3D05172005</td>
</tr>
</tbody>
</table>

Engine/Host Dependent Information

OID                   | IG.AE       |
Name                  | Adverse Events |
SASDatasetName        | AE           |
IsReferenceData        | Yes          |
Repeating             | Yes          |
Domain                | AE           |
Origin                |              |
Role                  |              |
Purpose               |              |
Comment               | Some adverse events from this trial |

List of Variables and Attributes

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Datatype</th>
<th>Length</th>
<th>SigDig</th>
<th>Format</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AECONTRT</td>
<td>string</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AEBCTRRT</td>
<td>string</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>AEBOUT</td>
<td>string</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AEREL</td>
<td>string</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>AESEV</td>
<td>string</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>AEENDT</td>
<td>date</td>
<td></td>
<td></td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>AEENYR</td>
<td>integer</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>AEENDAY</td>
<td>integer</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>AEENNMON</td>
<td>integer</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>AESSTD</td>
<td>date</td>
<td></td>
<td></td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>AESSTYR</td>
<td>integer</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>AESSTDAY</td>
<td>integer</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>AESSTMON</td>
<td>integer</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>AETERM</td>
<td>string</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>LINE_NO</td>
<td>integer</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>F_STATUS</td>
<td>string</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SCTRY</td>
<td>string</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>FNO</td>
<td>string</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>TAREA</td>
<td>string</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Program with LONGNAMES=YES

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

1. The FILENAME statement assigns the file reference Xmlinp to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.

2. The PROC CDISC statement specifies the following:
   - CDISC ODM as the model.
• File reference 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. Names 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. SAS variable names are captured from the Name= attribute in the ODM ItemDef elements. SAS format names are captured from the Name= attribute in the ODMCodeList elements.

4. The CONTENTS statement writes the contents of the CDISC ODM SAS data set in the SAS log. The CONTENTS statement must specify the name that is in the Name= attribute in the ODM ItemGroupDef element.

```sas
filename Xmlinp 'C:\XML\AE.xml';
proc cdisc model=odm read=Xmlinp;
odm odmversion="1.2" longnames=yes;
contents table="Adverse_Events";
run;
```
Log 4.3 Contents of CDISC ODM SAS Data Set with LONGNAMES=YES

Contents of CDISC ODM Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Adverse_Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Libref</td>
<td>XMLINP</td>
</tr>
<tr>
<td>ODM Version</td>
<td>1.2</td>
</tr>
<tr>
<td>AsOfDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>CreationDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>SourceSystem</td>
<td>SAS 9.1</td>
</tr>
<tr>
<td>SourceSystemVersion</td>
<td>9.01.01M3D05172005</td>
</tr>
</tbody>
</table>

Engine/Host Dependent Information

| OID                   | IG.AE          |
| Name                  | Adverse Events |
| SASDatasetName        | AE             |
| IsReferenceData       | Yes            |
| Domain                | AE             |
| Origin                |                |
| Role                  |                |
| Purpose               |                |
| Comment               | Some adverse events from this trial |

List of Variables and Attributes

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Datatype</th>
<th>Length</th>
<th>SigDig</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Actions_taken__other</td>
<td>string</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Actions_taken_re_study_drug</td>
<td>string</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Outcome</td>
<td>string</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Relationship_to_study__</td>
<td>string</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Severity</td>
<td>string</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Derived_Stop_Date</td>
<td>date</td>
<td></td>
<td>DATE</td>
</tr>
<tr>
<td>7</td>
<td>Stop_Year___Enter_Four_Digit_Yea</td>
<td>integer</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Stop_Day___Enter_Two_Digits_01_3</td>
<td>integer</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Stop_Month___Enter_Two_Digits_01</td>
<td>integer</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Derived_Start_Date</td>
<td>date</td>
<td></td>
<td>DATE</td>
</tr>
<tr>
<td>11</td>
<td>Start_Year___Enter_Four_Digit_Ye</td>
<td>integer</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Start_Day___Enter_Two_Digits_01</td>
<td>integer</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Start_Month___Enter_Two_Digits_0</td>
<td>integer</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Conmed_Indication</td>
<td>string</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Line_Number</td>
<td>integer</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Record_status__5_levels__internal</td>
<td>string</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Country</td>
<td>string</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Protocol_Number</td>
<td>string</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Therapeutic_Area</td>
<td>string</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Listing a Directory with the DATASETS Statement

Overview

This example produces a directory listing of a CDISC ODM XML document in the SAS log. 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 that illustrate how the ODM statement LONGNAMES= processing option determines the sources of captured SAS name parameters based on the value of the LONGNAMES= option.

To view the AE.XML document, see Appendix 1, “Sample CDISC ODM XML Document,” on page 93.

Program with LONGNAMES=NO

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

1. The FILENAME statement assigns the file reference Xmlinp to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.

2. The PROC CDISC statement specifies the following:
   - CDISC ODM as the model.
   - File reference 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. Names 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.

```sas
filename Xmlinp 'C:\XML\AE.xml';
proc cdisc model=odm read=Xmlinp;
odm odmversion="1.2" longnames=no;
datasets;
run;
```

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

```
Directory of CDISC ODM Tables

<table>
<thead>
<tr>
<th>Libref</th>
<th>XMLINP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODM Version</td>
<td>1.2</td>
</tr>
<tr>
<td>AsOfDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>CreationDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>SourceSystem</td>
<td>SAS 9.1</td>
</tr>
<tr>
<td>SourceSystemVersion</td>
<td>9.01.01M3D05172005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Tablename</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 AE</td>
</tr>
</tbody>
</table>
```
Program with `LONGNAMES=YES`

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

1. The `FILENAME` statement assigns the file reference `Xmlinp` to the physical location of the input XML document (complete pathname, filename, and file extension) to be imported.

2. The `PROC CDISC` statement specifies the following:
   - CDISC ODM as the model.
   - File reference `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. Names 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.

```sas
filename Xmlinp 'C:\XML\AE.xml';
proc cdisc model=odm read=Xmlinp;
   odm odmversion="1.2" longnames=yes;
   datasets;
run;
```

**Log 4.5  Directory Listing of the AE.XML Document in the SAS Log**

<table>
<thead>
<tr>
<th>Libref</th>
<th>XMLINP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODM Version</td>
<td>1.2</td>
</tr>
<tr>
<td>AsOfDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>CreationDateTime</td>
<td>2005-05-18T14:01:41</td>
</tr>
<tr>
<td>SourceSystem</td>
<td>SAS 9.1</td>
</tr>
<tr>
<td>SourceSystemVersion</td>
<td>9.01.01M3D05172005</td>
</tr>
</tbody>
</table>

```
# Tablename
  1 Adverse_Events
```
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.

```sas
libname Results 'C:\MyFiles\';
proc cdisc model=sdtm;
    sdtm sdtmversion="3.1";
    domaindata data=Results.AE domain=AE category=events;
run;

libname Results clear;
```
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;

proc cdisc model=sdtm;
  sdtm sdtmversion="3.1"
  domain data=Oralib.AE domain=AE category=events;
run;

libname Oralib clear;
Part 3

Procedure Reference

Chapter 6
CDISC ODM Procedure ...................................................... 59

Chapter 7
CDISC SDTM Procedure ..................................................... 85
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 for the procedure statements:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Task</th>
<th>Importing</th>
<th>Exporting</th>
</tr>
</thead>
<tbody>
<tr>
<td>“PROC CDISC for ODM Statement” on page 62</td>
<td>Specify CDISC ODM as the model and reference the physical location of the input or output XML document.</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>“ODM Statement” on page 64</td>
<td>Specify the CDISC ODM version and file type.</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>“STUDY Statement” on page 71</td>
<td>Specify the study identifier.</td>
<td>not valid</td>
<td>required</td>
</tr>
<tr>
<td>Statement</td>
<td>Task</td>
<td>Importing</td>
<td>Exporting</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>“GLOBALVARIABLES Statement” on page 72</td>
<td>Specify general summary information about the study.</td>
<td>not valid</td>
<td>required</td>
</tr>
<tr>
<td>“BASICDEFINITIONS Statement” on page 73</td>
<td>Specify information about measurement units that are used in the study.</td>
<td>not valid</td>
<td>optional</td>
</tr>
<tr>
<td>“METADATAVERSION Statement” on page 74</td>
<td>Specify the metadata version and version name that are used by the study.</td>
<td>not valid</td>
<td>required</td>
</tr>
<tr>
<td>“PRESENTATION Statement” on page 75</td>
<td>Specify information about how the study is presented to users.</td>
<td>not valid</td>
<td>optional</td>
</tr>
<tr>
<td>“USER Statement” on page 76</td>
<td>Specify information about users who are involved in the study.</td>
<td>not valid</td>
<td>optional</td>
</tr>
<tr>
<td>“LOCATION Statement” on page 78</td>
<td>Specify information about the physical location of the study.</td>
<td>not valid</td>
<td>optional</td>
</tr>
<tr>
<td>“SIGNATURE Statement” on page 80</td>
<td>Specify information about the signatures that are required for administering the study.</td>
<td>not valid</td>
<td>optional</td>
</tr>
<tr>
<td>“CLINICALDATA Statement” on page 81</td>
<td>When importing, identify the output SAS data set and specify where the data content in the XML document begins. When exporting, identify the input SAS data set and specify any optional metadata attributes.</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>“CONTENTS Statement” on page 83</td>
<td>Describe the contents of a CDISC ODM SAS data set in the SAS log.</td>
<td>optional</td>
<td>not valid</td>
</tr>
<tr>
<td>“DATASETS Statement” on page 83</td>
<td>Produce a directory listing of the CDISC ODM XML document in the SAS log.</td>
<td>optional</td>
<td>not valid</td>
</tr>
</tbody>
</table>

**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;
<BASICDEFINITIONS DATA=libref.member-name>;
METADATAVERSION metadata-attributes | DATA=libref.member-name;
<PRESENTATION DATA=libref.member-name>;
<User DATA=libref.member-name>;
<LOCATION DATA=libref.member-name>;
<SIGNATURE DATA=libref.member-name>;
CLINICALDATA OUT=libref.member-name
SASDATASETNAME="name" | DATA=libref.member-name <metadata-attributes>;
CONTENTS TABLE="name";
DATASETS;

<table>
<thead>
<tr>
<th>Statement</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC CDISC for ODM</td>
<td>Specify CDISC ODM as the model and reference the physical location of the input or output ODM XML document.</td>
</tr>
<tr>
<td>ODM</td>
<td>Specify the CDISC ODM version and file type.</td>
</tr>
<tr>
<td>STUDY</td>
<td>Specify the study identifier.</td>
</tr>
<tr>
<td>GLOBALVARIABLES</td>
<td>Specify general summary information about the study.</td>
</tr>
<tr>
<td>BASICDEFINITIONS</td>
<td>Specify information about measurement units that are used in the study.</td>
</tr>
<tr>
<td>METADATAVERSION</td>
<td>Specify the metadata version and version name that are used by the study.</td>
</tr>
<tr>
<td>PRESENTATION</td>
<td>Specify information about how the study is presented to users.</td>
</tr>
<tr>
<td>USER</td>
<td>Specify information about users who are involved in the study.</td>
</tr>
<tr>
<td>LOCATION</td>
<td>Specify information about the physical location of the study.</td>
</tr>
<tr>
<td>SIGNATURE</td>
<td>Specify information about the signatures that are required for administering the study.</td>
</tr>
<tr>
<td>CLINICALDATA</td>
<td>When importing, identify the output SAS data set and specify where the data content in the CDISC ODM XML document begins. When exporting, identify the input SAS data set and specify any optional metadata attributes.</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>Describe the contents of a CDISC ODM SAS data set in the SAS log.</td>
</tr>
</tbody>
</table>
### DATASETS
Produce a directory listing of the CDISC ODM XML document in the SAS log.

### 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.

**Example:** Chapter 4, “CDISC Procedure Examples for CDISC ODM,” on page 17

### Syntax
```
PROC CDISC MODEL=ODM READ=fileref | WRITE=fileref
<FORMATACTIVE=NO | YES> <FORMATNOREPLACE=NO | YES>
<FORMATLIBRARY=libref> <LANGUAGE=language-identifier>;
```

### 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 | WRITE=fileref**
specifies the SAS file reference that is assigned to the input or output XML document.

**Requirement** Required for both importing and 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 in the XML document to SAS formats 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 in the XML document to 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 that have the same names as the converted formats.

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

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

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 with the LANGUAGE= option 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-
ensure that it does not conflict with names that could be 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 23

ODM Statement

Specifies the CDISC ODM version 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 with the DATA= argument.

Example: Chapter 4, “CDISC Procedure Examples for CDISC ODM,” on page 17

Syntax

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

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 that is valid for SAS names.

NO

When importing, specifies that ODM name attributes are converted to SAS names. Names can be a maximum of eight characters in length. 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 name 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 of eight characters in length. 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= or SDSVarName= attribute 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 Appendix 1, “Sample CDISC ODM XML Document,” on page 93, 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 prior to Version 7, which includes Version 5 transport files.

**YES**
When importing, specifies that ODM name attributes are converted to SAS names. Names can be a maximum of 32 characters in length. PROC CDISC captures the SAS data set name from the Name= attribute in the ODM ItemGroupDef element. SAS variable names are captured from the Name= attribute in the ODM ItemDef elements. SAS format names are captured from the Name= attribute in the ODM 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. Names can be a maximum of 32 characters in length. PROC CDISC exports the SAS data set name to the Name= attribute in the ODM ItemGroupDef element. SAS variable names are exported to the Name= attribute in the ODM ItemDef elements, and SAS format names are exported to the Name= attribute in the ODM CodeList elements.

For example, by using the Appendix 1, “Sample CDISC ODM XML Document,” on page 93, 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 name 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 attributes are optional in CDISC ODM version 1.2. However, the Name= attribute for those ODM attributes 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 unique SubjectKey member is written to the output SAS data set.

When exporting, specifies that only the unique SubjectKey member 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**  
specifies 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.
<table>
<thead>
<tr>
<th>Alias</th>
<th>IGNORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>YES</td>
</tr>
<tr>
<td>Requirement</td>
<td>Optional for importing.</td>
</tr>
<tr>
<td>See</td>
<td>“Importing a CDISC ODM XML Document with OrderNumber Attributes” on page 27</td>
</tr>
</tbody>
</table>

**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. It imports 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. It exports 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= processing 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= processing 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 element 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 prior to 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.

**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 9.4). 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, 9.01.01MxPmmddyyyy). 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

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Importing</th>
<th>Exporting</th>
<th>Stored in SAS Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>LONGNAMES=NO</td>
<td>YES</td>
<td>optional</td>
<td>optional</td>
</tr>
<tr>
<td>ODMMAXIMUMOIDLENGTH=</td>
<td>optional</td>
<td>not valid</td>
<td>no</td>
</tr>
<tr>
<td>ODMMINIMUMKEYSET=NO</td>
<td>YES</td>
<td>optional</td>
<td>optional</td>
</tr>
<tr>
<td>ORDERNUMBER=YES</td>
<td>NO</td>
<td>optional</td>
<td>not valid</td>
</tr>
<tr>
<td>USENAMEASLABEL=NO</td>
<td>YES</td>
<td>optional</td>
<td>optional</td>
</tr>
<tr>
<td>ODMVERSION=</td>
<td>required</td>
<td>required</td>
<td>yes</td>
</tr>
<tr>
<td>FILEOID=</td>
<td>not valid</td>
<td>required</td>
<td>yes</td>
</tr>
</tbody>
</table>
STUDY Statement

Specifies the study identifier.

**Requirement:** Required for exporting.

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

**Examples:**
- “Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax” on page 29
- “Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets” on page 37
- “Exporting a CDISC ODM XML Document with Required and Optional Statements” on page 43

**Syntax**

```
STUDY metadata-attributes | DATA=libref.member-name;
```

**Arguments**

`metadata-attributes`

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 quotes.

---

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Importing</th>
<th>Exporting</th>
<th>Stored in SAS Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILETYPE=</td>
<td>not valid</td>
<td>required</td>
<td>yes</td>
</tr>
<tr>
<td>DESCRIPTION=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>GRANULARITY=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>ARCHIVAL=YES</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>CREATIONDATETIME=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>PRIORFILEOID=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>ASOFDATETIME=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>ORIGINATOR=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>SOURCESYSTEM=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
<tr>
<td>SOURCESYSTEMVERSION=</td>
<td>not valid</td>
<td>optional</td>
<td>yes</td>
</tr>
</tbody>
</table>
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.

**Examples:**
- “Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax” on page 29
- “Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets” on page 37
- “Exporting a CDISC ODM XML Document with Required and Optional Statements” on page 43

---

**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.

**Example:** “Exporting a CDISC ODM XML Document with Required and Optional Statements” on page 43

**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 could be standardized or registered with IAN.

**Requirement**  
Required.

```plaintext
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.

**Examples:**  
"Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax" on page 29

"Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets" on page 37

"Exporting a CDISC ODM XML Document with Required and Optional Statements" on page 43

---

### Syntax

```plaintext
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:

  ```plaintext
  METADATAVERSIONOID="name"
  ```

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

  **Requirement**  
  Required.

  ```plaintext
  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 METADATA VERSION 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.

**Example:** "Exporting a CDISC ODM XML Document with Required and Optional Statements" on page 43

## 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.

**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 could be 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 who are 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.

Example: “Exporting a CDISC ODM XML Document with Required and Optional Statements” on page 43

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 on 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.

Example: “Exporting a CDISC ODM XML Document with Required and Optional Statements” on page 43

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.
SIGNATURE Statement

Specifies information about the signatures that are required for administering the study.

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

**Requirements:**

- **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.

- **METHODOLOGY=DIGITAL | ELECTRONIC** specifies the form in which the signature was stored.

- **MEANING=** `"string"` specifies information about the context in which the signature has meaning. Enclose the string in single or double quotation marks.

- **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.

**Example:** “Exporting a CDISC ODM XML Document with Required and Optional Statements” on page 43

**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.

**Example:**

```
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.

**Example:**

```
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.

**Example:**

```
SIGNATURE DATA=libref.member-name;
```
**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.

**Examples:**
- "Importing a CDISC ODM XML Document Using Default KeySet Processing" on page 18
- "Importing a CDISC ODM XML Document Specifying KeySet Processing Options" on page 20
- "Importing a CDISC ODM XML Document Using a Language Identifier" on page 23
- "Importing a CDISC ODM XML Document with OrderNumber Attributes" on page 27
- "Exporting a CDISC ODM XML Document with Metadata Attributes in Statement Syntax" on page 29
- "Exporting a CDISC ODM XML Document with Metadata Attributes in SAS Data Sets" on page 37
- "Exporting a CDISC ODM XML Document with Required and Optional Statements" on page 43

**Syntax**

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

**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:

```markdown
NAME="string"
```

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

**Requirement**  Optional for exporting.

```markdown
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 `ODDMINIMUMKEYSET=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.
## CONTENTS Statement

Writes 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 64.

**Example:** “Describing a CDISC ODM SAS Data Set with the CONTENTS Statement” on page 46

### 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 64.

**Example:** “Listing a Directory with the DATASETS Statement” on page 50

### Syntax

```
DATASETS;
```
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>
<thead>
<tr>
<th>Statement</th>
<th>Task</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>“PROC CDISC for SDTM Statement” on page 86</td>
<td>Specify CDISC SDTM as the model and validate a SAS data set that conforms to CDISC SDTM.</td>
<td>yes</td>
</tr>
<tr>
<td>“SDTM Statement” on page 86</td>
<td>Specify the CDISC SDTM version number.</td>
<td>yes</td>
</tr>
<tr>
<td>“DOMAINDATA Statement” on page 87</td>
<td>Identify the SAS data set to be validated.</td>
<td>yes</td>
</tr>
</tbody>
</table>

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;

### Statement | Task
---|---
PROC CDISC for SDTM | Specify CDISC SDTM as the model and validate a SAS data set that conforms to CDISC SDTM.
SDTM | Specify the CDISC SDTM version number.
DOMAINDATA | Identify the SAS data set to be validated.

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

### Requirement: Required.

### Examples:
- "Validating CDISC SDTM Data in a SAS Data Set" on page 53
- "Validating CDISC SDTM Data in an Oracle Table" on page 54

## 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.

### Requirement: Required.

### Examples:
- "Validating CDISC SDTM Data in a SAS Data Set" on page 53
- "Validating CDISC SDTM Data in an Oracle Table" on page 54
Syntax

SDTM SDTMVersion="version-number";

Required Argument

SDTMVersion="version-number"

specifies the CDISC SDTM version. 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.

Examples:  "Validating CDISC SDTM Data in a SAS Data Set" on page 53
           "Validating CDISC SDTM Data in an Oracle Table" on page 54

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

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 includes 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

<table>
<thead>
<tr>
<th>Supported CDISC SDTM Domain</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demography</td>
<td>DM</td>
</tr>
</tbody>
</table>
The following table lists the supported model types:

<table>
<thead>
<tr>
<th>Domain</th>
<th>Model Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demography</td>
<td>Special</td>
</tr>
<tr>
<td>Comments</td>
<td>Special</td>
</tr>
<tr>
<td>Concomitant Medications</td>
<td>Interventions</td>
</tr>
<tr>
<td>Exposure</td>
<td>Interventions</td>
</tr>
<tr>
<td>Substance Use</td>
<td>Interventions</td>
</tr>
<tr>
<td>Adverse Events</td>
<td>Events</td>
</tr>
</tbody>
</table>

**CATEGORY=** model-type

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

Requirement Required.
<table>
<thead>
<tr>
<th>Domain</th>
<th>Model Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disposition</td>
<td>Events</td>
</tr>
<tr>
<td>Medical History</td>
<td>Events</td>
</tr>
<tr>
<td>ECG Test Results</td>
<td>Findings</td>
</tr>
<tr>
<td>Inclusion/Exclusion Exception</td>
<td>Findings</td>
</tr>
<tr>
<td>Laboratory Test Results</td>
<td>Findings</td>
</tr>
<tr>
<td>Physical Examinations</td>
<td>Findings</td>
</tr>
<tr>
<td>Questionnaires</td>
<td>Findings</td>
</tr>
<tr>
<td>Subject Characteristics</td>
<td>Findings</td>
</tr>
<tr>
<td>Vital Signs</td>
<td>Findings</td>
</tr>
</tbody>
</table>

Requirement: Required.
Part 4

Appendixes

Appendix 1

Sample CDISC ODM XML Document .......................... 93
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/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="Adverse events from the CTChicago file"
AsOfDateTime="2005-05-18T14:01:41"
CreationDateTime="2005-05-18T14:01:41"
SourceSystem="SAS 9.1"
SourceSystemVersion="9.01.01M3D05172005">
<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>
</Study OID="STUDY.StudyOID">
<!--
GlobalVariables is a REQUIRED section in ODM markup
-->
<GlobalVariables>
</GlobalVariables>
</ODM>
<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>

<!--
  Columns defined in the table
-->  
<ItemDef 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" Two Digits 01-12" DataType="integer" Length="2" />
<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.$AESEV" />
</ItemDef>
<CodeList OID="CL.$AEREL" SASFormatName="$AEREL" Name="$AEREL" DataType="text">
    <CodeListItem CodedValue='0'>None</CodeListItem>
    <CodeListItem CodedValue='1'>Unlikely</CodeListItem>
    <CodeListItem CodedValue='2'>Possible</CodeListItem>
    <CodeListItem CodedValue='3'>Probable</CodeListItem>
</CodeList>

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

<CodeList OID="CL.$AEACTTR" SASFormatName="$AEACTTR" Name="$AEACTTR" DataType="text">
    <CodeListItem CodedValue='0'>None</CodeListItem>
    <CodeListItem CodedValue='1'>Discontinued permanently</CodeListItem>
    <CodeListItem CodedValue='2'>Reduced</CodeListItem>
</CodeList>
Interrupted
None
Medication required
Hospitalization required or prolonged
Other

Administrative metadata

Clinical Data: AE
Adverse Events
Some adverse events from this trial

98 Appendix 1 • Sample CDISC ODM XML Document
<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>
Recommended Reading

Here is the recommended reading list for this title:

- SAS Companion that is specific to your operating environment
- Base SAS focus area at support.sas.com/rnd/base
- For information about XML (Extensible Markup Language), see the Web site www.w3.org/XML

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books
Index

A
accessibility features 4

B
BASICDEFINITIONS statement
CDISC procedure 73

C
CDISC models 3
CDISC ODM 5
   KeySet members 6
   metadata attributes 9
CDISC ODM examples
   describing data sets with CONTENTS statement 46
   exporting with metadata attributes in data sets 37
   exporting with metadata attributes in statement syntax 29
   exporting with optional statements 43
   importing with default KeySet processing 18
   importing with KeySet processing options 20
   importing with language identifiers 23
   importing with OrderNumber attributes 27
   listing a directory with DATASETS statement 50
CDISC ODM procedure
   syntax 60
CDISC overview 3
CDISC procedure 3
   syntax overview 4
CDISC SDTM 13
   validating data sets 13
CDISC SDTM examples
   validating data sets 53
   validating Oracle tables 54
CDISC STDM procedure
   syntax 85
CLINICALDATA statement
   CDISC procedure 81
   CONTENTS statement 46
   CDISC procedure 83

D
data sets
   describing with CONTENTS statement 46
   validating for CDISC SDTM 53
DATASETS statement 50
   CDISC procedure 83
   directory listing 50
DOMAINDATA statement
   CDISC procedure 87

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 43

G
GLOBALVARIABLES statement
   CDISC procedure 72

I
importing
   CDISC ODM, with default KeySet processing 18
   CDISC ODM, with KeySet processing options 20
   CDISC ODM, with language identifiers 23
   CDISC ODM, with OrderNumber attributes 27
K
KeySet members  6, 18, 20

L
language identifiers  23
LOCATION statement
  CDISC procedure  78

M
metadata attributes  29, 37
metadata attributes for CDISC ODM  9
METADATAVERSION statement
  CDISC procedure  74

O
ODM statement
  CDISC procedure  64
Oracle tables
    validating for CDISC SDTM  54
OrderNumber attributes  27

P
PRESENTATION statement
  CDISC procedure  75
PROC CDISC for ODM statement
  CDISC procedure  62
PROC CDISC for SDTM statement
  CDISC procedure  86

S
SDTM statement
  CDISC procedure  86
SIGNATURE statement
  CDISC procedure  80
statements (optional)  43
STUDY statement
  CDISC procedure  71

U
USER statement
  CDISC procedure  76

V
validating
  CDISC SDTM data sets  13, 53
  CDISC SDTM Oracle tables  54

X
XML documents  5