



SAS Publishing



# **The CDISC Procedure for SAS<sup>®</sup> Software**

Release 8.2 and Later

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *The CDISC Procedure for SAS® Software, Release 8.2 and Later*. Cary, NC: SAS Institute Inc.

**The CDISC Procedure for SAS® Software, Release 8.2 and Later**

Copyright © 2004, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

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

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

1st printing, July 2004

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at [support.sas.com/pubs](http://support.sas.com/pubs) or call 1-800-727-3228.

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

Other brand and product names are registered trademarks or trademarks of their respective companies.

---

# Contents

## **Chapter 1** $\triangle$ **The CDISC Procedure** 1

Overview 1

Understanding How ODM Markup Is Mapped to SAS Components 2

Specifying Import Statement Parameters 7

Specifying Export Statement Parameters 8

Syntax: CDISC Procedure 11

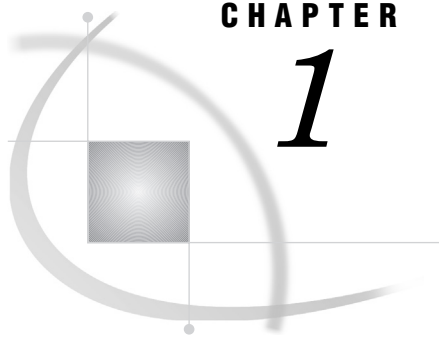
Examples 24

## **Appendix 1** $\triangle$ **Sample XML Document** 37

Sample CDISC ODM Document 37

**Index** 47





## CHAPTER

## 1

# The CDISC Procedure

<i>Overview</i>	1
<i>What Does the CDISC Procedure Do?</i>	1
<i>Importing an XML Document That Is in CDISC ODM Format</i>	2
<i>Exporting an XML Document in CDISC ODM Format</i>	2
<i>Understanding How ODM Markup Is Mapped to SAS Components</i>	2
<i>Understanding ODM Basics</i>	2
<i>How PROC CDISC Maps Elements and Attributes in an Import Operation</i>	3
<i>ODM Elements Created by a PROC CDISC Export Operation</i>	4
<i>Options for Handling Keyset Fields</i>	6
<i>Specifying Import Statement Parameters</i>	7
<i>Specifying Export Statement Parameters</i>	8
<i>Advantages of Specifying Statement Parameters in a Data Set</i>	10
<i>Syntax: CDISC Procedure</i>	11
<i>PROC CDISC Statement</i>	12
<i>ODM Statement</i>	14
<i>STUDY Statement</i>	15
<i>GLOBALVARIABLES Statement</i>	16
<i>METADATAVERSION Statement</i>	17
<i>CLINICALDATA Statement</i>	17
<i>BASICDEFINITIONS Statement</i>	19
<i>PRESENTATION Statement</i>	20
<i>USER Statement</i>	21
<i>LOCATION Statement</i>	22
<i>SIGNATURE Statement</i>	23
<i>Examples</i>	24
<i>Example 1: Importing a CDISC ODM Document Using Default Keyset Parameters</i>	24
<i>Example 2: Importing a CDISC ODM Document Specifying ODMMinimumKeyset=YES and ODMMaximumOIDLength</i>	26
<i>Example 3: Export That Specifies Parameters in the Statement Syntax</i>	28
<i>Example 4: Export That Specifies Parameters in a Data Set</i>	32
<i>Example 5: Export That Specifies Required and Optional Statements</i>	33

---

## Overview

---

### What Does the CDISC Procedure Do?

The CDISC procedure enables customers who are using Release 8.2 or later of SAS software to import and export XML documents that are in CDISC ODM 1.2 format.

CDISC is an acronym for “Clinical Data Interchange Standards Consortium.” ODM is an acronym for “Operational Data Model.”

ODM is one of many data models that are being defined by the standards consortium. It supports the acquisition, interchange, and archiving of clinical trials data for the medical and biopharmaceutical industries. Version 1.2 supports an XML schema representation of the ODM model.

---

## Importing an XML Document That Is in CDISC ODM Format

Importing an XML document that is in CDISC ODM format is the process of reading the external XML document as a SAS data set. The procedure translates the input XML document to the proprietary SAS file format.

To import a CDISC ODM document, execute PROC CDISC specifying the MODEL=, READ=, and optionally the FORMATACTIVE= and FORMATNOREPLACE= parameters, as well as the ODM and CLINICALDATA statements.

---

## Exporting an XML Document in CDISC ODM Format

Exporting an XML document in CDISC ODM format is the process of writing a SAS data set of type DATA to an output XML document that is in CDISC ODM 1.2 format. The procedure exports an XML document by translating the proprietary SAS file format to CDISC ODM XML markup.

To export a CDISC ODM document, execute PROC CDISC specifying the MODEL=, WRITE=, and optionally the FORMATACTIVE= parameter, as well as the following required and optional statements.

Required export statements:

- ODM
- STUDY
- GLOBALVARIABLES
- METADATAVERSION
- CLINICALDATA

Optional export statements:

- BASICDEFINITIONS
- PRESENTATION
- USER
- LOCATION
- SIGNATURE

---

## Understanding How ODM Markup Is Mapped to SAS Components

---

### Understanding ODM Basics

ODM defines the following entities to represent clinical study data:

item

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

item group

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

form

describes a set of logically or temporally related information.

study event

describes a patient visit or some other data collection event. A series of forms are collected as part of a study event. A study event is associated with a specific patient in the study.

The ODM model 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.

Each metadata entity contains internal and external attributes that identify the data entity for which it provides information. The internal attributes designate entities within the model and allow cross-references to be defined between entities both within and between ODM files. Internal attributes include an object instance identifier (OID), a subject key, and repeat keys.

- The 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 relates the distinct events to each other.

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

The set of attributes that are required in order to reference a single entity is referred to as the entity's *keyset*.

---

## How PROC CDISC Maps Elements and Attributes in an Import Operation

In an import operation, PROC CDISC creates a SAS data set, using version and keyset information that is specified in the ODM statement, and using data from the ItemGroupDef that is specified in the SASDATASETNAME= parameter of the CLINICALDATA statement.

ItemGroupDef attributes are mapped to SAS components as shown in Table 1.1 on page 4.

**Table 1.1** How PROC CDISC Resolves ODM Attributes in an Import Operation

ODM Attribute	Import Status	SAS Field
ItemGroupDef SASDatasetName= attribute	As is	SAS Data Set Name
ItemGroupDef Name= attribute	Convention corrected	
ItemGroupDef OID= attribute	Convention corrected	
ItemGroupDef Comment= attribute	As is	SAS Data Set Label
ItemGroupDef Name= attribute	As is	
ItemDef SASFieldName= attribute	As is	SAS Column Name
ItemDef Name= attribute	Convention corrected	
ItemDef OID= attribute	Convention corrected	
ItemDef Name= attribute	As is	SAS Column Label

For more information, see “Options for Handling Keyset Fields” on page 6 and “Specifying Import Statement Parameters” on page 7.

## ODM Elements Created by a PROC CDISC Export Operation

An export operation creates the following ODM elements:

- A SAS data set is represented as an ItemGroupDef.
- Each column in a data set is represented as an ItemRef in the appropriate ItemGroupDef, and a corresponding ItemDef is generated with the column metadata attributes.
 

*Note:* There is an ODM Normative Document non-conformance issue: The ODM-ND requires the SignificantDigits attribute (decimal precision) on an ItemDef for a floating point value (type=float). This is not currently generated.  $\Delta$
- If a column contains a reference to a user-defined SAS format, a CodeListRef is generated within the ItemDef of the column. The procedure generates a CodeList element for each unique user-defined SAS format that is referenced. A CodeListRef refers to a single CodeList element, whereas a CodeList element is referenced by multiple CodeListRefs.
- In addition, PROC CDISC automatically generates the following entities:
  - a FormDef containing an ItemGroupRef for each ItemGroupDef
  - a StudyEventDef containing a FormRef for each FormDef
  - a Protocol containing a StudyEventRef for each StudyEventDef
  - A CLINICALDATA element for each ItemGroupDef that is created.

All of the metadata is contained in a single METADATAVERSION element.

Keyset attributes are either read from the SAS data set or created by PROC CDISC, depending on the setting of the ODMMinimumKeySet= parameter and on whether keys have been defined in the SAS data set.



*Note:* Regardless of the ODMMinimumKeySet= setting, the input SAS data set must contain a unique subject key in each row.  $\Delta$

The following table shows how specific attribute values are assigned.

**Table 1.2** How PROC CDISC Assigns Attribute Values in an Export Operation

Element	Attribute	Value
ItemGroupDef	*OID	"IG." plus SAS data set member name
	*Repeating	"No"
	SASDataSetName	SAS data set member name
	*Name	from CLINICALDATA statement
	Domain	from CLINICALDATA statement
	Origin	from CLINICALDATA statement
	Role	not used per ODM-ND
	Purpose	from CLINICALDATA statement
	Comment	from CLINICALDATA statement
ItemRef	*ItemOID	"ID." plus SAS column name
	OrderNumber	generated sequence order
	*Mandatory	"No"
	KeySequence	ignored
	ImputationMethodOID	ignored
	Role	ignored
	RoleCodeListOID	ignored
ItemDef	*OID	"ID." plus SAS column name
	*Name	SAS column label
	SASFieldName	SAS column name
	*DataType	mapped from SAS data type to either "text" or "float"
	Length	maximum length of string type; ignored for non-string types
	SignificantDigits	ignored (see Note above)
	SDSVarName	ignored
	Origin	ignored
Comment	ignored	
CodeListRef	*CodeListOID	"CL." plus SAS format name

Element	Attribute	Value
CodeList	*OID	"CL." plus SAS format name
	*Name	SAS format name
	*DataType	"text" or "float"
	SASFormatName	SAS format name

\* Indicates a required parameter

For more information, see “Options for Handling Keyset Fields” on page 6 and “Specifying Export Statement Parameters” on page 8.

## Options for Handling Keyset Fields

ODM OIDs can have up to 100 characters each. Therefore, you might want to decrease the length of the keyset values if you are storing the values in the resulting data set, or you might want to store only the unique SubjectKey keyset value when importing from an ODM file.

PROC DISC provides the ODM parameters `ODMMinimumKeyset=` and `ODMMaximumOIDLength=` to enable you to specify how the keyset fields are handled by the import and export processes.

Here is an example of keyset fields from a CDISC ODM document:

```
<ClinicalData StudyOID="StudyOID" MetaDataVersionOID="v1.1.0">
  <SubjectData SubjectKey="001">
    <StudyEventData StudyEventOID="StudyEventOID" StudyEventRepeatKey="1">
      <FormData FormOID="AE" FormRepeatKey="1">
        <ItemGroupData ItemGroupOID="AE" ItemGroupRepeatKey="1" TransactionType="Insert">
```

The following table shows how the keyset fields are converted when a CDISC ODM document is imported into a SAS table, and how the fields are created when a SAS table is exported to a CDISC ODM document.

**Table 1.3** How Keyset Fields are Converted in an Import Operation and Created in an Export Operation

SAS Column Name	Keyset Field	How Created on Export
<code>__STUDYOID</code>	<code>StudyOID=</code>	User specifies via PROC CDISC statements
<code>__METADATAVERSIONOID</code>	<code>MetaDataVersionOID=</code>	User specifies via PROC CDISC statements
<code>__STUDYEVENTOID</code>	<code>StudyEventOID=</code>	User specifies via PROC CDISC statements
<code>__STUDYEVENTREPEATKEY</code>	<code>StudyEventRepeatKey=</code>	Automatically generated by the procedure
<code>__FORMOID</code>	<code>FormOID=</code>	User specifies via PROC CDISC statements
<code>__FORMREPEATKEY</code>	<code>FormRepeatKey=</code>	Automatically generated by the procedure

SAS Column Name	Keyset Field	How Created on Export
__ITEMGROUPOID	ItemGroupOID=	User specifies via PROC CDISC statements
__ITEMGROUPEPEATKEY	ItemGroupRepeatKey=	Automatically generated by the procedure
__TRANSACTIONTYPE	TransactionType=	Value "insert" is always used
__SUBJECTKEY	SubjectKey=	Read from the exported data set

When used in an import operation:

ODMMinimumKeyset=NO  
specifies that all keyset fields are present in the study data.

ODMMinimumKeyset=YES  
specifies to keep only the SubjectKey keyset field in the study data.

ODMMaximumOIDLength=*number*  
enables you to specify a smaller character length for the key fields. The default value is the maximum length defined in the ODM model.

When used in an export operation:

ODMMinimumKeyset=NO  
specifies to keep all keyset fields in the study data.

ODMMinimumKeyset=YES  
specifies that only the SubjectKey keyset field is present in the study data.

For an example of how the keyset options affect the data in an import operation, see “Examples” on page 24.

## Specifying Import Statement Parameters

When importing a CDISC ODM document, specify parameters for the ODM and CLINICALDATA statements as part of the statement syntax. For example:

```
FILENAME XMLINP 'an-ODM-document-to-import.xml' ;

PROC CDISC    MODEL=ODM
              READ=XMLINP
              formatActive=YES
              formatNoReplace=NO
              ;

ODM          ODMVersion="1.2"
              ODMmaximumOIDLength=16
              ODMminimumKeyset=NO
              ;

CLINICALDATA OUT=my.ae
              SASDATASETNAME=AE"
              ;
```

```
RUN;

FILENAME XMLINP;
```

This example imports data from an ItemGroupDef that has a SASDatasetName= attribute value of "AE". All ODM keyset fields are retained in the resulting MY.AE SAS data set. A maximum length of 16 characters is allocated to each keyset field.

---

## Specifying Export Statement Parameters

When exporting to a CDISC ODM document, you must follow these conventions:

- You can specify parameters for the ODM, STUDY, GLOBALVARIABLES, and METADATAVERSION statements in one of two ways:
  - directly in the statement syntax
  - stored in a SAS data set and referenced in the statement in a DATA= parameter.

Here is an example of an ODM statement that specifies export parameters as part of the statement syntax:

```
ODM      ODMVersion="1.2"
         FileOID="000--00--0000"
         FileType=SNAPSHOT
         Description="Adverse events from the CTChicago file";
```

Here is an example of referencing those same parameters in the DATA parameter:

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

In this example, CURRENT.ODM is a SAS data set that contains the ODM statement parameters that were shown in the previous example. The data set could have been created with the following code:

```
data current.odm;
  ODMVersion="1.2";
  FileOID="ODM.FileOID";
  FileType="SNAPSHOT";
  Description="Adverse events from the CTChicago file";
run;
```

- The required export parameters for the CLINICALDATA statement must be passed in a SAS data set that is specified in the DATA= parameter. Optional export parameters must be specified as part of the statement syntax. For example:

```
CLINICALDATA DATA=my.ae
             DOMAIN="AE"
             NAME="Adverse Events"
             COMMENT="All adverse events in this trial";
```

In this example, the SAS data set MY.AE contains the required statement parameters that are shown in Table 1.4 on page 9. DOMAIN=, NAME=, and COMMENT= are optional CLINICALDATA parameters.

- Parameters for the BASICDEFINITIONS, PRESENTATION, USER, LOCATION, and SIGNATURE statements must be passed in a SAS data set in the DATA= parameter.

The following tables list the required and optional statements for exporting a CDISC ODM document. The tables also summarize the required and optional parameters that are defined for each statement. For more information, see the descriptions of the individual statements.

**Table 1.4** Required Export Statements

Statement	Required Parameters	Optional Parameters
ODM	ODMVersion FileOID FileType	Description Granularity PriorFileOID Archival Originator AsOfDateTime CreationDateTime SourceSystem SourceSystemVersion
STUDY	StudyOID	
GLOBALVARIABLES	StudyName StudyDescription ProtocolName	
METADATAVERSION	MetaDataVersionOID Name	
CLINICALDATA	__StudyOID* __MetadataVersionOID* __SubjectKey __StudyEventOID* __FormOID* __ItemGroupOID* __ItemGroupRepeatKey* __TransactionType*	Name Domain Origin Purpose Comment

\* These keyset fields are required only if you specify ODMMinimumKeyset=NO.

**Table 1.5** Optional Export Statements

Statement	Required Parameters	Optional Parameters
BASICDEFINITIONS	MeasurementOID Name Lang TranslatedText	
PRESENTATION	PresentationOID	Lang TranslatedText

Statement	Required Parameters	Optional Parameters
USER	UserOID	UserType LoginName DisplayName FullName FirstName LastName Organization StreetName City StateProv Country PostalCode OtherText Email PictFileName PictImageType Pager Fax Phone LocationOID
LOCATION	LocationOID Name LocationType StudyOID MetaDataVersionOID EffectiveDate	
SIGNATURE	SignatureOID Methodology Meaning LegalReason	

## Advantages of Specifying Statement Parameters in a Data Set

Using the DATA= parameter instead of specifying statement parameters directly in the statement syntax for all statements enables you to use the same execution code for all export operations. It also enables you to change the actual metadata and data content by directing the LIBNAME specifications to different locations. For example, the following procedure code could export data for many studies:

```
LIBNAME metadata 'C:\your-meta-library';
LIBNAME clindata 'C:\your-data-library';
FILENAME XMLOUT '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;

FILENAME XMLOUT;

```

See “Examples” on page 24 for more information.

---

## Syntax: CDISC Procedure

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

---

**PROC CDISC MODEL=ODM**

**READ=***fileref* | **WRITE=***fileref*

<**FORMATACTIVE=** YES | NO <**FORMATNOREPLACE=** YES | NO>>;

**ODM** *parameters* | **DATA=***libref.member-name*;

**STUDY** *parameters* | **DATA=***libref.member-name*;

**GLOBALVARIABLES** *parameters* | **DATA=***libref.member-name*;

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

**METADATAVERSION** *parameters* | **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 DATA=libref.member-name <parameters>;**

To do this	Use these PROC CDISC parameters	And these statements
Import a CDISC ODM document	MODEL= READ= FORMATACTIVE= FORMATNOREPLACE=	ODM CLINICALDATA
Export a CDISC ODM document	MODEL= WRITE= FORMATACTIVE=	ODM STUDY GLOBALVARIABLES METADATAVERSION CLINICALDATA and optionally: BASICDEFINITIONS PRESENTATION USER LOCATION SIGNATURE

---

## PROC CDISC Statement

### Imports or exports an XML document in CDISC ODM format

**Requirement:** The PROC CDISC statement cannot be issued without related statements. When importing a CDISC ODM document, use the procedure with the ODM and CLINICALDATA statements.

When exporting a CDISC ODM document, use the procedure with the following required statements:

ODM  
STUDY  
GLOBALVARIABLES  
METADATAVERSION  
CLINICALDATA

You can also use the following optional statements:

BASICDEFINITIONS  
PRESENTATION  
USER  
LOCATION  
SIGNATURE

---



**PROC CDISC****MODEL=ODM****READ=fileref | WRITE=fileref****<FORMATACTIVE=YES | NO <FORMATNOREPLACE=YES | NO>>;****MODEL=ODM**

specifies the name of a supported CDISC model. In the current release, the valid value is *ODM*. This parameter is required for both import and export operations.

**READ=libref**

when importing an XML document in CDISC ODM format, specifies a SAS fileref that was assigned to the input XML document. This parameter is not supported in an export operation.

**WRITE=fileref**

when exporting an XML document in CDISC ODM format, specifies a SAS fileref that was assigned for the procedure output. This parameter is not supported in an import operation.

**FORMATACTIVE=YES | NO**

specifies whether CDISC ODM CodeList elements, which contain instructions for transcoding display data in a CDISC ODM document, are to be converted to SAS variable formats, and vice versa.

In an import operation, specifying **FORMATACTIVE=YES** converts the CDISC ODM CodeList elements to the corresponding SAS formats, registers the SAS formats on the referenced variables, and stores the created SAS formats in the SAS **FORMAT** library.

In an export operation, specifying **FORMATACTIVE=YES** converts SAS formats to the corresponding CDISC ODM CodeList elements.

In both import and export operations, specifying **FORMATACTIVE=NO** causes formats to be ignored.

**Default:** NO**FORMATNOREPLACE=YES | NO**

specifies whether to replace existing format entries in the format catalog search path in cases where an existing format entry has the same name as a format that is being created by PROC CDISC when it converts a CDISC ODM CodeList element.

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

When **FORMATNOREPLACE=NO**, the procedure replaces existing formats that have the same name.

**Default:** NO

**Restriction:** Use this option only when you are importing a CDISC ODM document, and only if the **FORMATACTIVE** option is set to **YES**.

**FORMATLIBRARY=libref**

specifies a permanent storage location for variable formats that are created by the procedure when **FORMATACTIVE=YES**.

---

## ODM Statement

### Specifies information about the ODM version and the file type

**Requirement:** The ODM statement is required for both importing and exporting a CDISC ODM document.

**Import Usage:** When importing a CDISC ODM document, use the ODM statement with the ODMVersion=, ODMMinimumKeyset=, and optionally, the ODMMaximumOIDLength= parameters.

**Export Usage:** When exporting a CDISC ODM document, use the ODM statement with the ODMVersion=, FileOID=, and FileType= parameters, and optionally the Description=, Granularity=, PriorFileOID=, Archival=, Originator=, AsOfDateTime=, CreationDateTime=, SourceSystem=, and SourceSystemVersion= parameters.

**General Usage:** Whether importing or exporting, you can either specify statement parameters in the statement syntax or store them in a SAS data set that you can reference in the DATA= parameter.

---

**ODM parameters** | DATA=libref.member-name;

## Required Parameters

### ODMVersion=*version-number*

specifies an ODM model number. The valid value is 1.2. This parameter is supported for importing and exporting.

### FileOID=*identifier*

specifies a unique identifier for the export file. This parameter is supported for exporting only.

### FileType=SNAPSHOT | TRANSACTIONAL

specifies the document's file type. This parameter is supported for exporting only.

#### SNAPSHOT

refers to a document that contains only the current state of the data and metadata that it describes, with no transactional history. A SNAPSHOT document supports one instruction per data point.

#### TRANSACTIONAL

refers to a document that contains the current state of the data and metadata, and that also includes the transactional history. A TRANSACTIONAL document supports more than one instruction per data point.

## Optional Parameters

### ODMMinimumKeyset=NO | YES

specifies whether to keep all ODM keyset fields that are in the study data. This parameter is supported for importing and exporting. NO specifies to keep all of the keyset fields. YES specifies to keep only the SubjectKey keyset field. The default value is NO. For more information about using the ODMMinimumKeyset= parameter, see “Options for Handling Keyset Fields” on page 6.

**ODMMaximumOIDLength=number**

enables you to specify a smaller character length for the ODM keyset fields. This parameter is supported for importing only. The default value is the maximum length defined in the ODM model.

**DATA=libref.member-name**

specifies a SAS data set that contains ODM statement parameters. Use the DATA= parameter if you want to submit statement parameters in a SAS data set instead of specifying them in the statement syntax. See “Specifying Export Statement Parameters” on page 8 and “Advantages of Specifying Statement Parameters in a Data Set” on page 10 for more information.

**Description=string**

specifies an optional text string that provides details to supplement the other attributes that are described in the document. This parameter is supported for exporting only.

**PriorFileOID=name**

specifies an optional reference to the previous file, if any, in a series. This parameter is supported for exporting only.

**Originator=name**

identifies the organization that generated the ODM file. This parameter is supported for exporting only.

**AsOfDateTime=datetime-value**

specifies the date and time when the source database was queried to create the document, in compliance with ISO-8601 guidelines. This parameter is supported for exporting only.

**CreationDateTime=datetime-value**

specifies the time when the file that contains the document was created or transmitted, in compliance with ISO-8601 guidelines. This parameter is supported for exporting only.

**SourceSystem=string**

specifies the application that created the file or transmission. This parameter is supported for exporting only. The default value is the short name of the current SAS release (for example, **SAS 9.1**).

**SourceSystemVersion=string**

specifies the version identifier of the application that created the file or transmission. This parameter is supported for exporting only. The default value is the annotated SAS release name (for example, **9.01.01MxPmmdyyyy**).

---

## STUDY Statement

**Specifies the study identifier**

**Requirement:** The STUDY statement is required for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

**Usage:** Statement parameters can either be specified in the statement syntax or stored in a SAS data set and submitted in the DATA= parameter.

---

**STUDY parameters** | DATA=libref.member-name;

## Required Statement Parameter

### **StudyOID=ODM-identifier**

specifies a unique study identifier. A string of characters up to the ODM maximum defined length is supported.

## Optional Parameter

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

specifies a SAS data set that contains STUDY statement parameters. Use the DATA= parameter if you want to submit statement parameters in a SAS data set instead of specifying them in the statement syntax. See “Specifying Export Statement Parameters” on page 8 and “Advantages of Specifying Statement Parameters in a Data Set” on page 10 for more information.

---

## GLOBALVARIABLES Statement

**Specifies general, summary information about the study**

**Requirement:** The GLOBALVARIABLES statement is required for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

**Usage:** Statement parameters can either be specified in the statement syntax or stored in a data set and submitted in the DATA= parameter.

---

**GLOBALVARIABLES** *parameters* | **DATA=libref.member-name**;

## Required Statement Parameters

### **StudyName=name**

specifies the short external name of the study.

### **StudyDescription=string**

specifies a description of the study.

### **ProtocolName=name**

specifies the sponsor’s internal name for the protocol.

## Optional Statement Parameter

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

specifies a SAS data set that contains GLOBALVARIABLES statement parameters. Use the DATA= parameter if you want to submit statement parameters in a SAS data set instead of specifying them in the statement syntax. See “Specifying Export Statement Parameters” on page 8 and “Advantages of Specifying Statement Parameters in a Data Set” on page 10 for more information.

---

## METADATAVERSION Statement

**Specifies a metadata version**

**Requirement:** The METADATAVERSION statement is required for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

**Usage:** Statement parameters can either be specified in the statement syntax or stored in a data set and specified in the DATA= parameter. All required statements must use the same input method.

---

**METADATAVERSION** *parameters* | **DATA=***libref.member-name*;

### Required Statement Parameters

**MetadataVersionOID=***name*

specifies a metadata version identifier.

**Name=***name*

specifies a name for the metadata version.

### Optional Statement Parameters

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

specifies a SAS data set that contains METADATAVERSION statement parameters. Use the DATA= parameter if you want to submit statement parameters in a SAS data set instead of specifying them in the statement syntax. See “Specifying Export Statement Parameters” on page 8 and “Advantages of Specifying Statement Parameters in a Data Set” on page 10 for more information.

---

## CLINICALDATA Statement

**When used for importing a CDISC ODM document, specifies details for writing the SAS data set. When used for exporting a CDISC ODM document, identifies where data begins in the XML document.**

**Requirements:**

The CLINICALDATA statement is required for both importing and exporting a CDISC ODM document.

The statement supports different parameters for importing and exporting.

When importing a CDISC ODM document, specify statement parameters in the statement syntax.

When exporting to CDISC ODM format,

- store required statement parameters in a SAS data set that contains clinical study data, and reference the data set in the DATA= parameter.
  - specify optional statement parameters as part of the statement syntax.
-

**CLINICALDATA**

**OUT**=*libref.member-name* **SASDATASETNAME**="name";  
 | **DATA**=*libref.member-name* <*optional-parameters*>;

**Import Parameters**

**IN**=*libref.member-name*

specifies a library and member name for the resulting SAS data set.

**SASDATASETNAME**=*name*

specifies an ODM ItemGroupDef= value that identifies where the data content in the CDISC ODM document begins.

**Export Parameters**

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

specifies a SAS data set that contains the clinical study data as well as the following required CLINICALDATA metadata parameters.

\_\_**StudyOID**=*identifier*

specifies a unique identifier for the study. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**MetadataVersionOID**=*identifier*

specifies the metadata version that is used by the study. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**SubjectKey**=*key-value*

specifies a subject within the study. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**StudyEventOID**=*identifier*

specifies a StudyEventDef in the study. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**FormOID**=*identifier*

specifies a FormDef in the study. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**ItemGroupOID**=*identifier*

specifies an ItemGroup in the study. This is a required parameter and must be included in a SAS data set in the DATA= parameter along with the clinical data. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**ItemGroupRepeatKey**=*key-value*

specifies an item group repeat key. A string of characters up to the maximum length defined in the ODM model is supported.

\_\_**TransactionType**=INSERT | UPDATE | REMOVE | UPSERT | CONTEXT

specifies the transaction type.

**INSERT**

specifies that the data entity is new and must be added to the study along with the properties provided. An error is returned if the entity already exists.

**UPDATE**

specifies that the data entity already exists and must be modified in order to have new properties added. Existing properties are not modified. An error is returned if the entity does not exist.

**REMOVE**

specifies that the data entity exists and must be deleted along with all of its properties and its children. An error is returned if the entity does not exist.

**UPSERT**

specifies to modify the specified entity if it exists or to create the entity if it does not exist.

**CONTEXT**

specifies that the data is being re-sent for context purposes only.

**Name=string**

specifies the study name. This is an optional parameter.

**Domain=domain-name**

**Origin=string**

**Purpose=string**

**Comment=string**

specify submission information as defined in the CDISC Submission Metadata Model. These are optional parameters and can be specified only in the CLINICALDATA statement.

---

## BASICDEFINITIONS Statement

**Specifies information about the measurement units that were used in the study**

**Reminder:**

The BASICDEFINITIONS statement is optional for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

Statement parameters for this statement must be stored in a SAS data set and submitted to the statement in the DATA= parameter.

All statement parameters are required.

---

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

**DATA=libref.member-name**

specifies a SAS data set that contains the following required BASICDEFINITIONS statement parameters:

**MeasurementOID=ODM-identifier**

specifies a symbol or abbreviation that represents a measurement unit. A string of characters up to the maximum length defined by the ODM model is supported.

**Name=name**

specifies the name of the measurement unit.

**Lang=language-identifier**

specifies a language identifier. As defined in the XML specification, a language identifier can be any 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); these begin with the prefix "i-" or "I-".
- a language identifier that has been assigned by the user or agreed on between parties in private use; these must begin with the prefix "x-" or "X-" in order to ensure that they do not conflict with names that are later standardized or registered with IANA.

**TranslatedText=string**

specifies the name of the measurement unit in the specified language.

---

## PRESENTATION Statement

**Specifies information about how the study is presented to users**

**Restriction:**

The PRESENTATION statement is optional for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document..

Statement parameters for this statement must be stored in a SAS data set and submitted to the statement in the DATA= parameter.

---

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

### Required Statement Parameters

**DATA=libref.member-name**

specifies a SAS data set that contains the required and optional PRESENTATION statement parameters.

**PresentationOID=ODM-identifier**

specifies a reference to a presentation definition. A string of characters up to the maximum length defined in the ODM model is supported.

### Optional Statement Parameters

**Lang=language-identifier**

specifies a language identifier. As defined in the XML specification, a language identifier can be any 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); these begin with the prefix "i-" or "I-".



- a language identifier that has been assigned by the user or agreed on between parties in private use; these must begin with the prefix "x-" or "X-" in order to ensure that they do not conflict with names that are later standardized or registered with IANA.

**TranslatedText=string**

specifies an example of the presentation in the specified language.

---

## USER Statement

**Specifies information about a user of a clinical data collection system, such as an investigator or a data management staff member**

**Restriction:**

The USER statement is optional for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

Statement parameters must be stored in a SAS data set and submitted to the statement in the DATA= parameter.

---

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

### Required Statement Parameters

**DATA=libref.member-name**

specifies a SAS data set that contains required and optional USER statement parameters.

**UserOID=ODM-identifier**

specifies an identifying reference to a system user. A string of characters up to the maximum length defined by the ODM model is supported.

### Optional Statement Parameters

**UserType=SPONSOR | INVESTIGATOR | LAB | OTHER**

specifies the named user's role in the study.

**LoginName=user-ID**

specifies the user ID that the named user uses to log in to the clinical study data system.

**DisplayName=name**

specifies a short name for the user.

**FullName=name**

specifies the full name of the user.

**LastName=name**

specifies the last name of the user.

**FirstName=name**

specifies the first name of the user.

**Organization=*name***

specifies the user's organization.

**StreetName=*street-address***

specifies the street address of the user's postal address.

**City=*name***

specifies the city name in the user's postal address.

**StateProv=*state-or-province***

specifies the state or province in the user's postal address.

**Country=*name***

specifies the country name in the user's postal address. This value must be represented by an ISO 3166 two-letter country code.

**PostalCode=*code***

specifies the postal code in the user's postal address.

**OtherText=*string***

specifies any other text that is needed in the user's postal address. A string of characters up to the maximum length defined in the ODM model is supported.

**Email=*e-mail-address***

specifies the user's e-mail address.

**PictFileName=*filename***

specifies a filename that contains a picture of the user.

**PictImageType=*file-type***

specifies the image file type.

**Pager=*number***

specifies the user's pager number.

**Fax=*number***

specifies the user's fax number.

**Phone=*number***

specifies the user's phone number.

**LocationOID=*ODM-identifier***

specifies a reference to a location definition. A string of characters up to the maximum length defined in the ODM model is supported.

---

## LOCATION Statement

**Specifies information about a physical location****Restriction:**

The LOCATION statement is optional for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

Statement parameters must be stored in a SAS data set and submitted to the statement in the DATA= parameter.

All statement parameters are required.

---

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

**DATA=libref.member-name**

specifies a SAS data set that contains the following LOCATION statement parameters:

LocationOID=*ODM-identifier*

specifies a unique identifier for a location. A string of characters up to the maximum length defined in the ODM model is supported.

Name=*location-name*

specifies the name of the location.

LocationType=SPONSOR | SITE | CRO | LAB | OTHER

specifies the type of location.

StudyOID=*ODM-identifier*

specifies a unique study identifier in which this location is participating. A string of characters up to the maximum length defined in the ODM model is supported.

MetaDataVersionOID=*ODM-identifier*

specifies the metadata version that is used at the location. A string of characters up to the maximum length defined in the ODM model is supported.

EffectiveDate=*date*

specifies the date of the metadata version, in compliance with ISO-8601 guidelines.

---

## SIGNATURE Statement

**Specifies information about the signatures that are stored in the CDISC ODM document**

**Restrictions:**

The SIGNATURE statement is optional for exporting a CDISC ODM document. It is not supported for importing a CDISC ODM document.

Statement parameters must be stored in a SAS data set and submitted to the statement in the DATA= parameter.

All statement parameters are required.

---

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

**DATA=libref.member-name**

specifies a SAS data set that contains the following statement parameters:

SignatureOID=*ODM-identifier*

specifies a unique identifier for the signature. A string of characters up to the maximum length defined in the ODM model 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.

LegalReason=*string*

specifies why signature authentication is necessary. A string of characters up to the maximum length defined in the ODM model is supported.

---

## Examples

---

### Example 1: Importing a CDISC ODM Document Using Default Keyset Parameters

#### Procedure features:

- This example imports a CDISC ODM document named AE.XML and creates a SAS data set named RESULTS.AE. “AE” is the ItemGroupDef SASDatasetName= attribute value in the XML source document.
- Specifying ODMMinimumKeyset=NO in the ODM statement (which is also the default value if the ODMMinimumKeyset= parameter is omitted from the procedure) causes all key fields to be stored in the resulting data set. By default, the maximum length that is defined in the ODM model is allocated to each key.

**Source file:** To view AE.XML, see “Sample CDISC ODM Document” on page 37.

---

#### Program

```
LIBNAME results '.' ;

FILENAME XMLINP 'ae.xml';

PROC CDISC          MODEL=ODM
                   READ=XMLINP

                   formatActive=YES
                   formatNoReplace=NO
                   ;
ODM                 ODMVersion = "1.2"
                   ODMminimumKeyset=NO
                   ;

CLINICALDATA       OUT = results.AE
                   SASDATASETNAME = "AE"
                   ;

RUN;
FILENAME XMLINP;
```

#### Sample Output

The following procedure prints the contents of RESULTS.AE, which are shown below.

```
proc print data=results.AE; run;
```

**Example Code 1.1** PROC PRINT Output for RESULTS.AE

```

                                The SAS System

              ___Meta
              Data      ___
Obs ___StudyOID  VersionOID  Subject  ___Study  ___StudyEvent  ___Form
                                Key      EventOID  RepeatKey  OID      ___FormRepeatKey

1  123-456-789   v1.1.0    001    SE.VISIT1                FORM.AE
2  123-456-789   v1.1.0    001    SE.VISIT1                FORM.AE

              ___Item
              Group  Transaction
Obs ___Item  GroupOID  RepeatKey  Type  TAREA      PNO      SCTRY      F_STATUS

1  IG.AE      1      Insert  Oncology  143-02  United States  Source verified, queried
2  IG.AE      2      Insert  Oncology  143-02  United States  Source verified, queried

Obs LINE_NO  AETERM  AESTMON  AESTDAY  AESTYR  AESTDT  AEENMON  AEENDAY  AEENYR  AEENDT

1    1    HEADACHE  06      10      1999   19990610  06      14      1999   19990614
2    2    CONGESTION 06      11      1999   19990611

Obs AESEV  AEREL  AEOUT      AEACTION  AEACTION  AEACTION  AEACTION  AEACTION  AEACTION  AEACTION

1  Mild  None  Resolved, no residual effects  None  Medication required
2  Mild  None  Continuing  None  Medication required

```

The output from PROC CONTENTS displays the file's attributes as well as the attributes of each interpreted column (variable), such as the variable's type and length. The attributes are obtained from the embedded ODM metadata content. The VARNUM option causes the variables to be printed in the order of their creation.

```

title 'DEFAULT BEHAVIOR';
proc contents data=results.AE varnum; run;

```

**Example Code 1.2** PROC CONTENTS Output for RESULTS.AE

```

                                DEFAULT BEHAVIOR

                                The CONTENTS Procedure

Data Set Name      RESULTS.AE  Observations      2
Member Type       DATA      Variables         29
Engine            XML        Indexes           0
Created           .        Observation Length 33
Last Modified     .        Deleted Observations 0
Protection        .        Compressed        NO
Data Set Type     .        Sorted            NO

```

```

Label
Data Representation Default
Encoding             Default

```

## Variables in Creation Order

#	Variable	Type	Len	Format	Label
1	__StudyOID	Char	100		
2	__MetaDataVersionOID	Char	100		
3	__SubjectKey	Char	100		
4	__StudyEventOID	Char	100		
5	__StudyEventRepeatKey	Char	100		
6	__FormOID	Char	100		
7	__FormRepeatKey	Char	100		
8	__ItemGroupOID	Char	100		
9	__ItemGroupRepeatKey	Char	100		
10	__TransactionType	Char	100		
11	TAREA	Char	4	\$TAREAF.	Therapeutic Area
12	PNO	Char	15		Protocol Number
13	SCTRY	Char	4	\$SCTRYF.	Country
14	F_STATUS	Char	1	\$F_STATU.	Record status,5 levels, internal use
15	LINE_NO	Num	8		Line Number
16	AETERM	Char	100		Conmed Indication
17	AESTMON	Char	2		Start Month-Enter 2 Digits 01-12
18	AESTDAY	Char	2		Start Day-Enter 2 Digits 01-31
19	AESTYR	Char	4		Start Year-Enter 4 Digit Year
20	AESTDT	Char	8		Derived Start Date
21	AEENMON	Char	2		Stop Month-Enter 2 Digits 01-12
22	AEENDAY	Char	2		Stop Day-Enter 2 Digits 01-31
23	AEENYR	Char	4		Stop Year-Enter 4 Digit Year
24	AEENDT	Char	8		Derived Stop Date
25	AESEV	Char	1	\$AESEV.	Severity
26	AEREL	Char	1	\$AEREL.	Relationship to study drug
27	AEOUT	Char	1	\$AEOUT.	Outcome
28	AEACTTRT	Char	1	\$AEACTTR.	Actions taken re study drug
29	AECONTRT	Char	1	\$AECONTR.	Actions taken, other

---

## Example 2: Importing a CDISC ODM Document Specifying ODMMinimumKeyset=YES and ODMMaximumOIDLength

**Procedure features:**

- This example imports a CDISC ODM document named AE.XML and creates a SAS data set named RESULTS.AEmin. “AE” is the ItemGroupDef SASDatasetName= attribute value in the XML source document.
- Specifying ODMMinimumKeyset=YES in the ODM statement causes only the Subject Key keyset field to be stored in the resulting SAS data set. Specifying ODMMaximumOIDLength=16 in the ODM statement allocates a storage space of

16 characters for the key value, instead of the default maximum length defined in the ODM model. For more information about the `ODMMinimumKeyset=` and `ODMMaximumOIDLength=` parameters, see “Options for Handling Keyset Fields” on page 6.

---

## Program

```
LIBNAME results  '.' ;

FILENAME XMLINP  'ae.xml';

PROC CDISC      MODEL=ODM
                READ=XMLINP
                formatActive=YES
                formatNoReplace=NO
                ;

ODM            ODMVersion = "1.2"
                ODMMinimumKeyset=YES
                ODMMaximumOIDLength=16
                ;

CLINICALDATA  OUT = results.AEmin
                SASDATASETNAME = "AE"
                ;

RUN;

FILENAME XMLINP ;
```

## Sample Output

The output from PROC CONTENTS displays the file’s attributes as well as the attributes of each interpreted column (variable), such as the variable’s type and length. The VARNUM option causes the variables to be printed first in the order of their creation.

```
title 'ODMMinimumKeyset=YES ODMMaximumOIDLength=16';
proc contents data=results.AEmin varnum; run;
```

### Example Code 1.3 PROC CONTENTS Output for RESULTS.AEmin

```
ODMMinimumKeyset=YES ODMMaximumOIDLength=16
```

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Label
1	__SubjectKey	Char	16	
2	TAREA	Char	4	Therapeutic Area
3	PNO	Char	15	Protocol Number
4	SCTRY	Char	4	Country

5	F_STATUS	Char	1	Record status, 5 levels, internal use
6	LINE_NO	Num	8	Line Number
7	AETERM	Char	100	Conmed Indication
8	AESTMON	Char	2	Start Month - Enter Two Digits 01-12
9	AESTDAY	Char	2	Start Day - Enter Two Digits 01-31
10	AESTYR	Char	4	Start Year - Enter Four Digit Year
11	AESTDT	Char	8	Derived Start Date
12	AEENMON	Char	2	Stop Month - Enter Two Digits 01-12
13	AEENDAY	Char	2	Stop Day - Enter Two Digits 01-31
14	AEENYR	Char	4	Stop Year - Enter Four Digit Year
15	AEENDT	Char	8	Derived Stop Date
16	AESEV	Char	1	Severity
17	AEREL	Char	1	Relationship to study drug
18	AEOUT	Char	1	Outcome
19	AEACTTRT	Char	1	Actions taken re study drug
20	AECONTRT	Char	1	Actions taken, other

---

### Example 3: Export That Specifies Parameters in the Statement Syntax

#### Procedure features:

This example specifies required export statements only.

Statement parameters are specified as part of the statement syntax. As an alternative, you can store parameters for the ODM, STUDY, GLOBALVARIABLES, METADATAVERSION, and optional export statements in separate SAS data sets and reference the data sets in a DATA= parameter. For an example, see Example 4 on page 32.

Source data set: RESULTS.AE

---

#### Program

```
LIBNAME results '.' ;

FILENAME XMLOUT 'AEfull.XML';

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 ;

```

## Sample Output

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

### Example Code 1.4 Output XML Document Generated by PROC CDISC

```

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

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

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

  ODMVersion="1.2"
  .
  .
  .
  SourceSystem="SAS 9.1"
  SourceSystemVersion="9.01.01MOD11182003">

<Study OID="Study.StudyOID">
  .
  .
  .
  8
<MetaDataVersion OID="v1.1.0" Name="Version 1.1.0">
  7

```

```

<Protocol>
  <StudyEventRef StudyEventOID="SE.VISIT1"
                    OrderNumber="1" Mandatory="Yes" />
</Protocol>
6
<StudyEventDef OID="SE.VISIT1"
                Name="Study Event Definition"
                Repeating="No" Type="Common">
  <FormRef FormOID="FORM.AE" OrderNumber="1" Mandatory="No" />
</StudyEventDef>
5
<FormDef OID="FORM.AE" Name="Form Definition" Repeating="No">
  <ItemGroupRef ItemGroupOID="IG.AE" Mandatory="No" />
</FormDef>
1
<ItemGroupDef OID="IG.AE" Repeating="No"
               SASDatasetName="AE"
               Name="Adverse Events"
               Domain="AE"
               Comment="All adverse events in this trial">
2
  <ItemRef ItemOID="ID.PNO" OrderNumber="1" Mandatory="No" />
  <ItemRef ItemOID="ID.AESEV" OrderNumber="2" Mandatory="No" />
  .
  .
  .
</ItemGroupDef>

<ItemDef OID="ID.PNO"
          SASFieldName="PNO"
          Name="Protocol Number"
          DataType="text" Length="15" />
<ItemDef OID="ID.AESEV"
          SASFieldName="AESEV"
          Name="Severity"
          DataType="text" Length="1">
3
  <CodeListRef CodeListOID="CL.$AESEV" />
</ItemDef>
.
.
.
4
<CodeList OID="CL.$AESEV"
           SASFormatName="$AESEV"
           Name="$AESEV"
           DataType="text">
  <CodeListItem CodedValue='1'>
    <Decode>
      <TranslatedText xml:Lang="en">Mild</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue='2'>

```

```

    <Decode>
      <TranslatedText xml:Lang="en">Moderate</TranslatedText>
    </Decode>
  </CodeListItem>
<CodeListItem CodedValue='3'>
  <Decode>
    <TranslatedText xml:Lang="en">Severe</TranslatedText>
  </Decode>
</CodeListItem>
<CodeListItem CodedValue='4'>
  <Decode>
    <TranslatedText xml:Lang="en">Life-threatening</TranslatedText>
  </Decode>
</CodeListItem>
</CodeList>
.
.
.
</MetaDataVersion>
</Study>
.
.
.
⑨
<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"
          TransactionType="Insert">
          <ItemData ItemOID="ID.PNO" Value="143-02" />
          <ItemData ItemOID="ID.AESEV" Value="1" />
          .
          .
          .
        </ItemGroupData>
        <ItemGroupData ItemGroupOID="IG.AE"
          ItemGroupRepeatKey="2"
          TransactionType="Insert">
          <ItemData ItemOID="ID.PNO" Value="143-02" />
          <ItemData ItemOID="ID.AESEV" Value="1" />
          .
          .
          .
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>

```

① A SAS data set is represented as an ItemGroupDef.

② Each column in the data set is generated as an ItemDef. Information about two columns is shown here.

- ③ If a column contains a reference to a user-generated SAS format, a CodeListRef is generated within the ItemDef of the column.
- ④ A CodeList is generated for each unique user-generated SAS format that is referenced.
- ⑤ A FormDef containing an ItemGroupRef for each ItemGroupDef is automatically generated.
- ⑥ A StudyEventDef containing a FormRef for each FormDef is automatically generated.
- ⑦ A Protocol containing a StudyEventRef for each StudyEventDef is automatically generated.
- ⑧ All of the metadata is contained by a single MetaDataVersion element.
- ⑨ A ClinicalData element is created for each ItemGroupDef that is referenced.

---

## Example 4: Export That Specifies Parameters in a Data Set

### Procedure features:

This example specifies required export statements only.

Statement parameters are stored in SAS data sets and are submitted to the statements in a DATA= parameter. For information about the advantages of using this method, see “Advantages of Specifying Statement Parameters in a Data Set” on page 10.

**Source data set:** RESULTS.AE

---

### Program

First, create data sets that contain statement parameters as follows:

```
LIBNAME current '.';

data current.odm;
  ODMversion = "1.2";
  FileOID = "ODM.FileOID";
  FileType = "Snapshot";
  Description = "Optional descriptive information";
run;

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

data current.globals;
  StudyName = "CDISC Test Study III";
  StudyDescription = "This file contains test data for CDISC testing";
  ProtocolName = "CDISC-Protocol-00-0000";
run;

data current.metadata;
  MetaDataVersionOID = "v1.1.0";
  Name = "Version 1.1.0";
run;
```

Then reference the data sets in PROC CDISC as follows:

```

FILENAME XMLOUT '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"
                COMMENT  = "All adverse events in this trial";

RUN;

FILENAME XMLOUT;

```

---

## Example 5: Export That Specifies Required and Optional Statements

### Procedure features:

This procedure executes optional export statements in addition to the required export statements.

Optional export statements must have parameters submitted in a SAS data set that is referenced in the DATA= parameter.

**Source data set:** RESULTS.AE

---

### Program

First, create data sets that contain the optional statement parameters:

```

libname current '.';

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

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

  MeasurementOID="MU.LB";

```

```

        Name="Pound";
        Lang="en";
        TranslatedText="English: Pound";
output;

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

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

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

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

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

data current.user;
LENGTH usertype $20.;
LENGTH organization $40.;
    UserOID="USR.cdisc001";
    UserType="Other";
    FullName="Fred Flintstone";
    FirstName="Fred";
    LastName="Flintstone";
    Organization = "CDISC";
    LocationOID="LOC.CDISCHome";
    StreetName="123 Main Street";

```

```

    City="Washington";
    StateProv="DC";
    Country="United States";
    PostalCode="";
output;

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

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

```

Then reference the data sets in PROC CDISC as follows:

```

FILENAME XMLOUT '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;

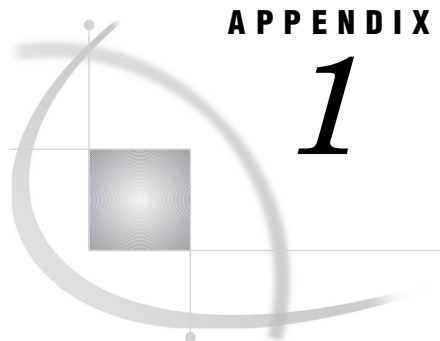
SIGNATURE           data=current.signature;

CLINICALDATA        data=current.ae
                   DOMAIN="AE"

```

```
NAME = "Adverse Events"  
COMMENT = "Adverse Events in the Clinical Trial";  
  
RUN;  
  
FILENAME XMLOUT;
```





## APPENDIX

## 1

## Sample XML Document

*Sample CDISC ODM Document 37*

### Sample CDISC ODM Document

Here is an example of an XML document that is in CDISC ODM format. This file is used in Example 1 on page 24.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!--
  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="2004-04-14T18:09:09"
  CreationDateTime="2004-04-14T18:09:09"
  SourceSystem="SAS 9.1"
  SourceSystemVersion="9.01.01M0D11182003">

<Study OID="123-456-789">

  <!--
    GlobalVariables is a REQUIRED section in ODM markup
  -->
  <GlobalVariables>
    <StudyName>CDISC Connect-A-Thon Test Study III</StudyName>
    <StudyDescription>This file contains test data for the CDISC
```

```

                                Connect-A-Thon event scheduled for the DIA 38th
                                annual meeting in Chicago.</StudyDescription>
    <ProtocolName>CDISC-Protocol-00-000</ProtocolName>
</GlobalVariables>

<BasicDefinitions />

<!--
    Internal ODM markup required metadata
    This section is generated from the data records contained
    in the CLINICALDATA table(s).
    Each table becomes its own form definition.
-->
<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="No" Type="Common">
    <FormRef FormOID="FORM.AE" OrderNumber="1" Mandatory="No" />
  </StudyEventDef>

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

  <!--
    Columns defined in the table
  -->
  <ItemGroupDef OID="IG.AE" Repeating="No"
    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" />
  </ItemGroupDef>

```

```

    <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"
    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="float" />

<ItemDef OID="ID.AETERM"
    SASFieldName="AETERM"
    Name="Conmed Indication"
    DataType="text" Length="100" />

<ItemDef OID="ID.AESTMON"
    SASFieldName="AESTMON"
    Name="Start Month - Enter Two Digits 01-12"
    DataType="text" Length="2" />

<ItemDef OID="ID.AESTDAY"
    SASFieldName="AESTDAY"
    Name="Start Day - Enter Two Digits 01-31"
    DataType="text" Length="2" />

```

```
<ItemDef OID="ID.AESTYR"  
  SASFieldName="AESTYR"  
  Name="Start Year - Enter Four Digit Year"  
  DataType="text" Length="4" />  
  
<ItemDef OID="ID.AESTDT"  
  SASFieldName="AESTDT"  
  Name="Derived Start Date"  
  DataType="text" Length="8" />  
  
<ItemDef OID="ID.AEENMON"  
  SASFieldName="AEENMON"  
  Name="Stop Month - Enter Two Digits 01-12"  
  DataType="text" Length="2" />  
  
<ItemDef OID="ID.AEENDAY"  
  SASFieldName="AEENDAY"  
  Name="Stop Day - Enter Two Digits 01-31"  
  DataType="text" Length="2" />  
  
<ItemDef OID="ID.AEENYR"  
  SASFieldName="AEENYR"  
  Name="Stop Year - Enter Four Digit Year"  
  DataType="text" Length="4" />  
  
<ItemDef OID="ID.AEENDT"  
  SASFieldName="AEENDT"  
  Name="Derived Stop Date"  
  DataType="text" Length="8" />  
  
<ItemDef OID="ID.AESEV"  
  SASFieldName="AESEV"  
  Name="Severity"  
  DataType="text" Length="1">  
  <CodeListRef CodeListOID="CL.$AESEV" />  
</ItemDef>  
  
<ItemDef OID="ID.AEREL"  
  SASFieldName="AEREL"  
  Name="Relationship to study drug"  
  DataType="text" Length="1">  
  <CodeListRef CodeListOID="CL.$AEREL" />  
</ItemDef>  
  
<ItemDef OID="ID.AEOUT"  
  SASFieldName="AEOUT"  
  Name="Outcome"  
  DataType="text" Length="1">  
  <CodeListRef CodeListOID="CL.$AEOUT" />  
</ItemDef>  
  
<ItemDef OID="ID.AEACTTRT"  
  SASFieldName="AEACTTRT"
```

```

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

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

<!--
    Translation to ODM markup for any PROC FORMAT style
    user formatting applied to columns in the table
-->

<CodeList OID="CL.$TAREAF"
    SASFormatName="$TAREAF"
    Name="$TAREAF"
    DataType="text">
    <CodeListItem CodedValue='ONC'>
        <Decode>
            <TranslatedText xml:lang="en">Oncology</TranslatedText>
        </Decode>
    </CodeListItem>
</CodeList>

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

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

```

```

        xml:lang="en">Source verified, queried</TranslatedText>
    </Decode>
</CodeListItem>
</CodeList>

```

```

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

```

```

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

```

```

    </Decode>
  </CodeListItem>
</CodeList>

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

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

```

```

        <TranslatedText xml:lang="en">Interrupted</TranslatedText>
    </Decode>
</CodeListItem>
</CodeList>

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

</MetaDataVersion>
</Study>

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

<!--
    Clinical Data    : AE
                    Adverse Events
                    All adverse events in this trial
-->
<ClinicalData StudyOID="123-456-789" MetaDataVersionOID="v1.1.0">
    <SubjectData SubjectKey="001">
        <StudyEventData StudyEventOID="SE.VISIT1">
            <FormData FormOID="FORM.AE">
                <ItemGroupData ItemGroupOID="IG.AE"
                    ItemGroupRepeatKey="1"

```



```

        TransactionType="Insert">
        <ItemData ItemOID="ID.TAREA" Value="ONC" />
        <ItemData ItemOID="ID.PNO" Value="143-02" />
        <ItemData ItemOID="ID.SCTRY" Value="USA" />
        <ItemData ItemOID="ID.F_STATUS" Value="V" />
        <ItemData ItemOID="ID.LINE_NO" Value="1" />
        <ItemData ItemOID="ID.AETERM" Value="HEADACHE" />
        <ItemData ItemOID="ID.AESTMON" Value="06" />
        <ItemData ItemOID="ID.AESTDAY" Value="10" />
        <ItemData ItemOID="ID.AESTYR" Value="1999" />
        <ItemData ItemOID="ID.AESTDT" Value="19990610" />
        <ItemData ItemOID="ID.AEENMON" Value="06" />
        <ItemData ItemOID="ID.AEENDAY" Value="14" />
        <ItemData ItemOID="ID.AEENYR" Value="1999" />
        <ItemData ItemOID="ID.AEENDT" Value="19990614" />
        <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"
        TransactionType="Insert">
        <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="19990611" />
        <ItemData ItemOID="ID.AEENMON" Value="" />
        <ItemData ItemOID="ID.AEENDAY" Value="" />
        <ItemData ItemOID="ID.AEENYR" Value="" />
        <ItemData ItemOID="ID.AEENDT" Value="" />
        <ItemData ItemOID="ID.AESEV" Value="1" />
        <ItemData ItemOID="ID.AEREL" Value="0" />
        <ItemData ItemOID="ID.AEOUT" Value="2" />
        <ItemData ItemOID="ID.AEACTTRT" Value="0" />
        <ItemData ItemOID="ID.AECONTRT" Value="1" />
    </ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>

```



# Your Turn

---

If you have comments or suggestions about *The CDISC Procedure for SAS Software, Release 8.2 and Later*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing  
SAS Campus Drive  
Cary, NC 27513  
E-mail: [yourturn@sas.com](mailto:yourturn@sas.com)

For suggestions about the software, please return the photocopy to

SAS Institute Inc.  
Technical Support Division  
SAS Campus Drive  
Cary, NC 27513  
E-mail: [suggest@sas.com](mailto:suggest@sas.com)