

Define.xml - Tips and Techniques for Creating CRT - DDS

Julie Maddox, SAS Institute Inc., Cary, NC, USA
Mark Lambrecht, SAS Institute Inc., Tervuren, Belgium

ABSTRACT

The Case Report Tabulation Data Definition Specification (also known as define.xml) provides the means by which submission metadata can be transmitted to regulatory agencies in a machine-readable XML format. Constructing this file can be a time-consuming process. This paper will show how centralized management of clinical data standards is integral to the creation of the CRT-DDS define.xml file. This paper will cover collection and extraction of metadata, customization of the CRT-DDS file, and tools to validate the CRT-DDS file. We will highlight best practices using SAS® Clinical Data Integration, a solution that focuses on pharmaceutical industry needs for transforming, managing, and verifying that data is in compliance with industry mandated data standards.

INTRODUCTION

The Case Report Tabulation Data Definition Specification is an XML format used for submission to a regulatory authority. This machine-readable file provides a list of the domains included in the submission, a description of each domain, and documentation on how the study data was collected. To assist the regulatory review process, the information in this file should be as detailed and complete as possible. SAS® Clinical Standards Toolkit and SAS Clinical Data Integration provide a practical solution for collecting the necessary metadata, as well as processes for customizing and validating the CRT-DDS file. This paper presents details on the tools SAS provides and specific examples of how to use these tools to produce the CRT-DDS format.

USING SAS CLINICAL DATA INTEGRATION TO CREATE THE CRT-DDS FILE

SAS Clinical Data Integration focuses on pharmaceutical industry needs for transforming, managing, and verifying industry mandated data standards such as those created by Clinical Data Interchange Standards Consortium (CDISC). This product relies upon the SAS® Metadata Server to provide centralized metadata management and SAS® Data Integration to provide tools to transform data. SAS Clinical Data Integration enhances usability by adding metadata objects and wizards which assist in collecting metadata on studies, domains, and domain columns. SAS Clinical Data Integration also provides a specialized transformation object for extracting metadata and creating the CRT-DDS XML format. This section describes how to use the functionality provided in SAS Clinical Data Integration to manage study and domain metadata and to create the CRT-DDS define.xml file.

MANAGING STUDY METADATA

Clinical Data Integration provides a wizard to create a new study or submission metadata object. The wizard will collect basic object metadata such as name, description, and content location in the SAS metadata tree. Next, metadata about the study is collected including the protocol phase and title, versions of data standards to be used in this study and which controlled terminology is available for the study. The study metadata can be easily maintained in the Study Properties window shown in Figure 1 below.

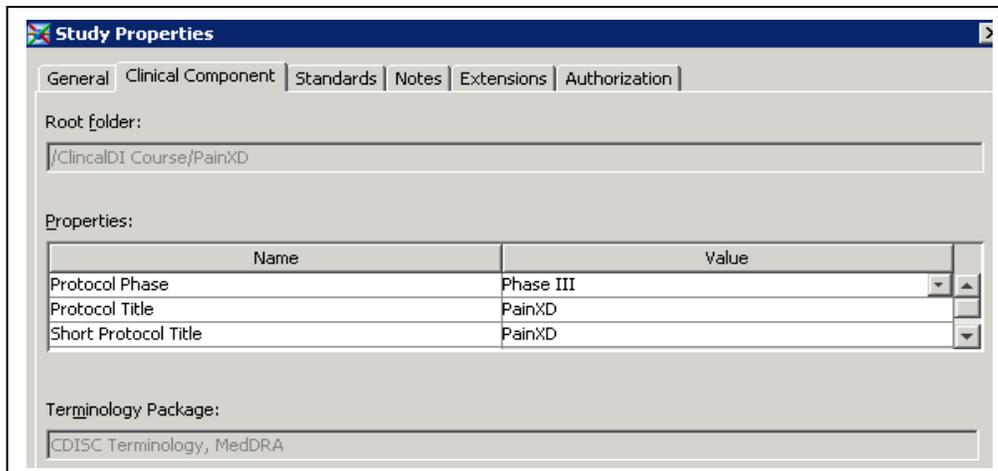


Figure 1 – Clinical Study Properties

PhUSE 2010

This study metadata feeds other SAS Clinical Data Integration plug-ins to ensure that users are consistently using the proper versions of standards for a given study. The Clinical Component metadata is used to populate the Study and CodeLists portion of the CRT-DDS file.

Managing Domain Metadata

Clinical Data Integration also collects clinical metadata on each CDISC SDTM domain and domain columns. For a domain, metadata on the Archive file, Class, Structure, and Purpose are defined as shown in the Clinical tab of the domain properties in Figure 2 below.

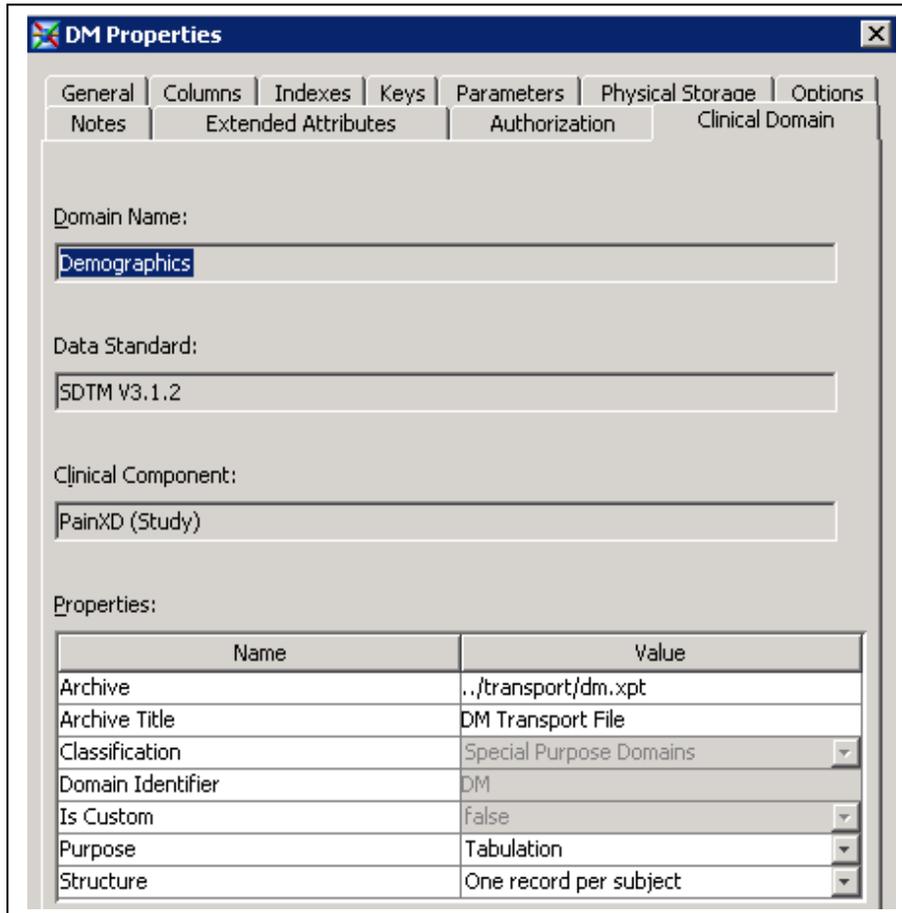


Figure 2 – Clinical Domain Properties

The domain metadata is primarily used to populate the ItemGroupDefs portion of the CRT-DDS file. In addition, clinical metadata is collected for each of the domain columns. This metadata can be maintained in the Clinical Column tab of the Column properties dialog (Figure 3). The clinical column metadata is primarily used to populate the ItemDefs portion of the CRT-DDS file.

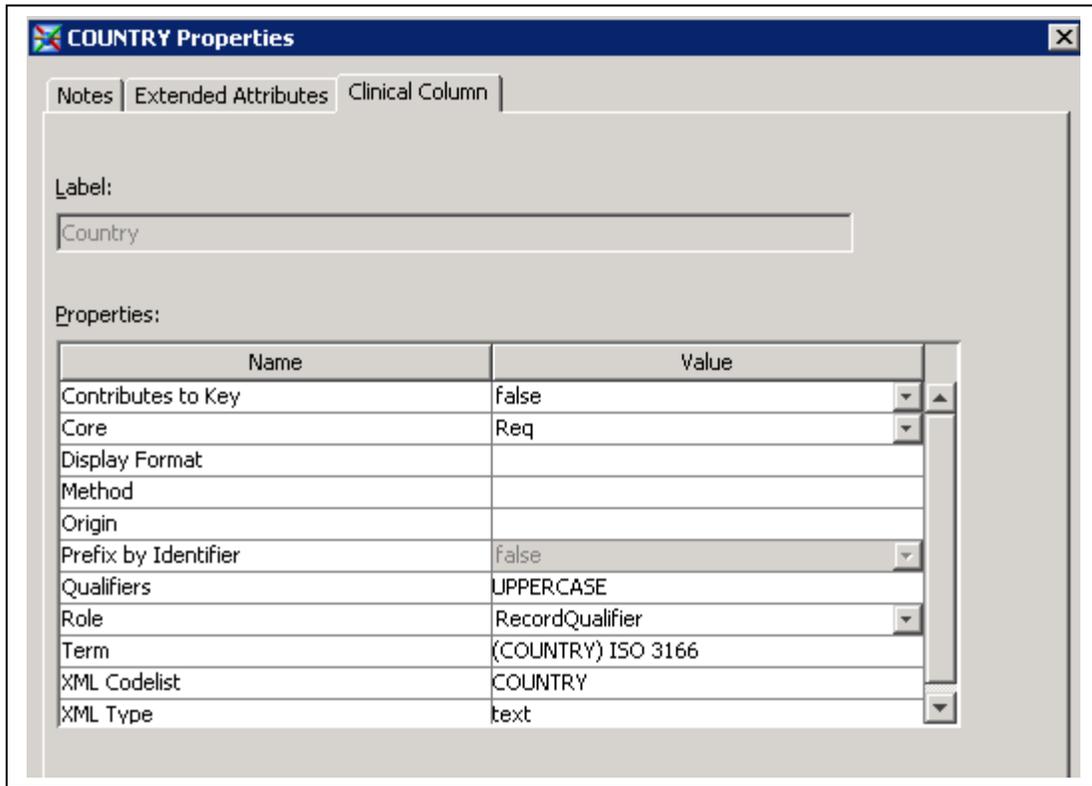


Figure 3 – Clinical Column Properties

CDISC-SDTM to CRT-DDS Transform

The metadata managed by SAS Clinical Data Integration can be published to CRT-DDS using the CDISC-SDTM to CRT-DDS Transform (Figure 4).

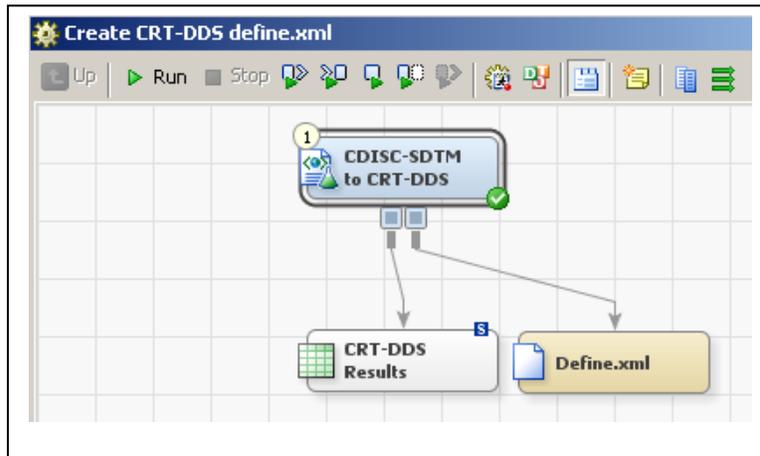


Figure 4 – Create CRT-DDS Define.xml Process Flow

This transform extracts metadata on the study, domains, domain columns and controlled terminology and generates SAS Clinical Standards Toolkit code to create the define.xml file. The transform properties (Figure 5) allow you to specify:

- which studies and domains to include in the define file
- output encoding
- customized style sheets
- a location to save the SAS CRT-DDS datasets

PhUSE 2010

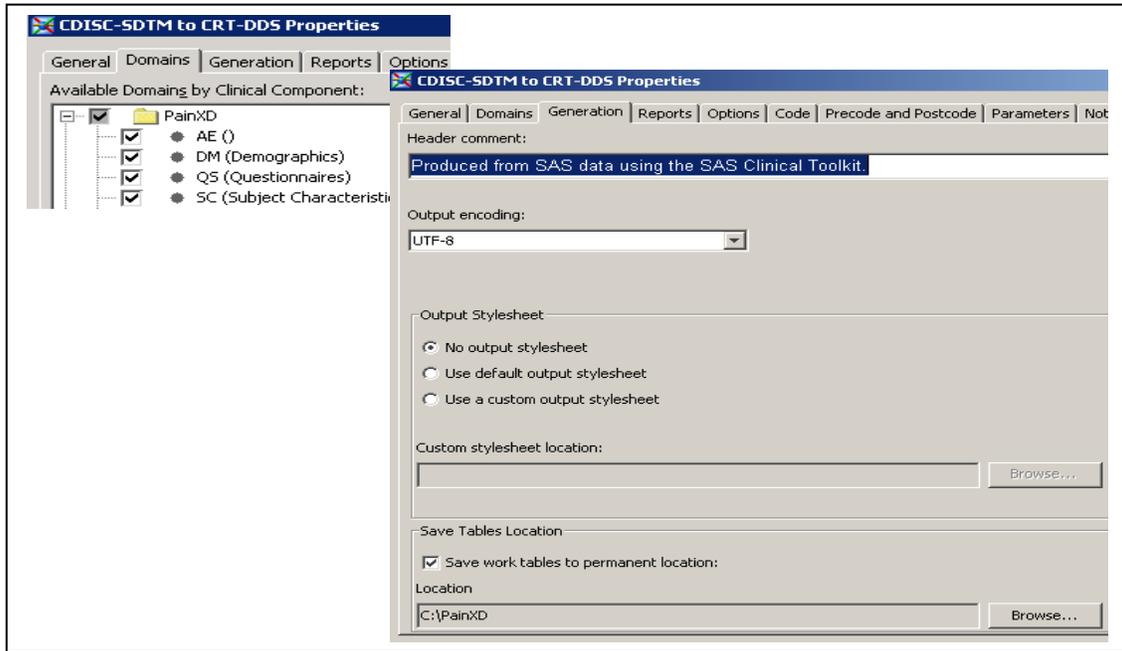


Figure 5 -CDISC-SDTM to CRT-DDS Transform Properties

When this job is executed, a define.xml file is produced and the SAS datasets representing the CRT-DDS model are created in the specified location. Note, the CDISC-SDTM to CRT-DDS transform, saves the SAS datasets used to create the define.xml file to the libref `_svWork`.

SUPPLYING ADDITIONAL METADATA FOR THE CRT-DDS FILE

To facilitate the process of creating a define.xml file, SAS provides a data model that represents the CRT-DDS Version 1.0 format. In general, the SAS representation of the CRT-DDS standard is patterned to match the XML element and attribute structure of the define.xml file. For example, domain-level metadata is represented by an ItemGroupDef element and is captured in the ItemGroupDefs SAS data set. This is illustrated in the following XML(Figure 6) and SAS dataset structure(Figure 7) below for the DM domain metadata.

```
<ItemGroupDef OID="DM1" Name="DM" Repeating="No" IsReferenceData="No"
  SASDatasetName="DM" Domain="DM" Purpose="Tabulation"
  Comment="additional information/notes on the DM domain"
  def:Label="Demographics" def:Class="Special Purpose"
  def:Structure="One record per subject"
  def:DomainKeys="STUDYID USUBJID"
  def:ArchiveLocationID="DM1">
  <def:leaf ID="DM1" xlink:href="../transport/dm.xpt">
    <def:title>DM transport file</def:title>
  </def:leaf>
</ItemGroupDef>
```

Figure 6 – Sample of ItemGroupDef element in CRT-DDS file

XML Tag	SAS Variable	Required?
OID	OID	Yes
Name	Name	Yes
Repeating	Repeating	Yes
IsReferenceData	IsReferenceData	No
SASDatasetName	SASDatasetName	No
Domain	Domain	No
Origin	Origin	No
Role	Role	No
Purpose	Purpose	No
Comment	Comment	No
def:Label	Label	Yes

PhUSE 2010

def:Class	Class	No
def:Structure	Structure	No
def:DomainKeys	DomainKeys	No
def:ArchiveLocationID	ArchiveLocationID	Yes

Figure 7 - SAS Representation of ItemGroupDefs Dataset

The information for def:leaf is placed in a separate ItemGroupLeaf table. A brief description of the SAS CRT-DDS datasets is presented in Appendix A. The data model details are provided as a reference for those users who wish to customize the CRT-DDS output. SAS Clinical Data Integration will create the necessary tables and execute the appropriate macros to produce the standard CRT-DDS file.

Customizing the CRT-DDS File

Once the SAS datasets representing the CRT-DDS model have been created, users can add metadata directly to these datasets, and then build a new define.xml file reflecting the updated metadata. In the example below we will demonstrate how to add an annotated case report form as well as value level metadata to the define.xml file. The CRT-DDS specification includes information which describes the forms used to collect the study data. Below is a description of the content of the SAS CRT-DDS datasets involved:

- AnnotatedCRF** – contains document references to the annotated case report form
- MDVLeaf** – contains the href link for each of the documents listed in the AnnotatedCRFs dataset
- MDVLeafTitles** – contains a descriptive title for each MDVLeaf item.

For this sample, we will demonstrate how to customize the CRT-DDS file inside a SAS Clinical Data Integration job. Add a User Written Code transform to the job flow (Figure 8), and modify the control order of the transformations so that the User Written Code transform runs after the CDISC-SDTM to CRT-DDS transform.

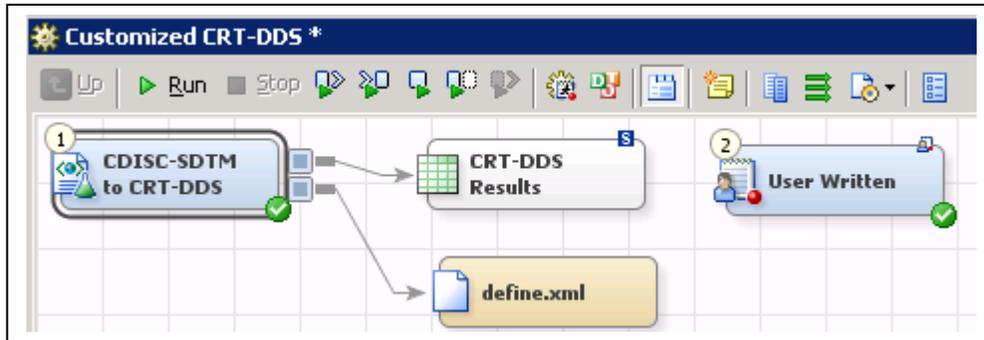


Figure 8 – Customized CRT-DDS Process Flow

Open the properties of the User Written transform and in the Code tab, select All user written for the Code Generation mode. (Figure 9)



Figure 9 – User Written Properties

In the code editor enter the following code (Figure 10), which will populate the AnnotatedCRF, MDVLeaf and MDVLeafTitles datasets.

PhUSE 2010

```

*Lookup OID for the SDTM 3.1.2 standard in MetaDataVersion;
proc sql noprint;
    select OID into :mdv from _svWork.MetaDataVersion
    where name="CDISC-SDTM 3.1.2";
quit;
*Add records for Annotated CRF;
proc sql;
    insert into _svWork.AnnotatedCRFs
    set    DocumentRef = "BlankCRF",
          leafID= "AnnotatedCRF",
          FK_MetaDataVersion = "&mdv";
    insert into _svWork.MDVLeaf
    set    ID= "AnnotatedCRF",
          href = "./blankcrf.pdf",
          FK_MetaDataVersion = "&mdv";
    insert into _svWork.MDVLeafTitles
    set    title= "Blank Annotated CRF",
          FK_MDVLeaf = "AnnotatedCRF";
quit;

```

Figure 10 – Sample Proc SQL code for AnnotatedCRF, MDVLeaf, MDVLeafTitles

The code in Figure 10 first looks up the unique object identifier (OID) for the data standard used to build the domains(CDISC-SDTM 3.1.2) in the MetaDataVersion dataset and places the value of OID into a macro variable for use in subsequent steps as a foreign key value. The next step adds a record to the AnnotatedCRFs dataset with the primary key leafID='AnnotatedCRF'. The next step adds a record to the MDVLeaf dataset with foreign key values which match the record added to the AnnotatedCRFs dataset and the MetaDataVersion dataset, and defines the href value for the PDF file. The last step adds a record to the MDVLeafTitles dataset which associates the PDF file in the MDVLeaf dataset with a descriptive title. Based on these updates to the SAS CRT-DDS datasets, Figure 11 displays the information which appears in the define.xml file.

```

<def:AnnotatedCRF>
  <def:DocumentRef
    leafID="AnnotatedCRF">BlankCRF</def:DocumentRef>
</def:AnnotatedCRF>
<def:leaf ID="AnnotatedCRF" xlink:href="./blankcrf.pdf">
  <def:title>Blank Annotated CRF</def:title>
</def:leaf>

```

Figure 11 – Partial define.xml file showing Annotated CRF metadata

Adding on to this example, we are defining value level metadata for the SCTEST variable in the Subject Characteristics (SC) domain. SDTM tabulation datasets have a normalized data structure (e.g., one record per patient per test code per visit), thus the structure of these datasets do not change when new information is added. Additional value level metadata is valuable to aid the reviewers. Below is a description of the content of the SAS CRT-DDS datasets involved:

- ValueLists** – contains id of value lists
- ValueListItemRefs**– contains id of each item in a value list
- ItemValueListRefs** – associates each value list item to a row in the ItemDefs dataset
- ItemDefs** – contains metadata for each id in ValueListItemRefs

For the SC domain in this sample study, Figure 12 shows the metadata at the data value level for the SCTEST column.

Column	Value	Label	Type	Length	Comment
SCTEST	HEIGHT	Height in inches	Text	10	Height taken barefoot
SCTEST	WEIGHT	Weight in pounds	Text	10	Weight taken without shoes

Figure 12 – Value level metadata for SCTEST column

To accomplish this task, add the code shown in Figure 13 to the code editor of the User Written Transform. This code will add records to the ValueLists, ItemValueListRefs, ItemDefs, and ValueListItemRefs datasets.

PhUSE 2010

```
*Lookup OID for the SCTEST column in ItemDefs;
proc sql noprint;
    select OID into :srccol from _svWork.ItemDefs
    where name='SCTEST';
quit;
*add record for a new valuelist SCTESTVALS;
proc sql ;
    insert into _svWork.ValueLists
    set  OID= "SCTESTVALS",
        FK_MetaDataVersion = "&mdv";
*add record associating the value list SCTESTVALS to the
OID for SCTEST ItemDefs record;
    insert into _svWork.ItemValueListRefs
    set  ValueListOID= "SCTESTVALS",
        FK_ItemDefs = "&srccol";
*add records to the ItemDefs dataset for each value in
the SCTESTVAL value list;
    insert into _svWork.ItemDefs
    set  OID= "VAL001",
        Name = "Height",
        DataType = "integer",
        Length = 3,
        SASFieldName = "HEIGHT",
        comment = "Height taken barefoot",
        label="Height in inches",
        FK_MetaDataVersion = "&mdv"
    set  OID= "VAL002",
        Name = "Weight",
        DataType = "integer",
        Length = 4,
        SASFieldName = "WEIGHT",
        comment = "Weight without shoes",
        label="Weight in pounds",
        FK_MetaDataVersion = "&mdv";
*add records associating the value list SCTESTVALS to
rows in the ItemDefs dataset;
    insert into _svWork.ValueListItemRefs
    set  ItemOID= "VAL001",
        OrderNumber=1,
        Mandatory="Yes",
        KeySequence=1,
        FK_ValueLists = "SCTESTVALS"
    set  ItemOID= "VAL002",
        OrderNumber=2,
        Mandatory="Yes",
        KeySequence=2,
        FK_ValueLists = "SCTESTVALS";
quit;
```

Figure 13 – Sample Proc SQL code for ValueLists, ItemValueListRefs, ItemDefs, and ValueListItemRefs

The code above first looks up the unique object identifier (OID) for the SCTEST column in the ItemDefs dataset and places the value of OID into a macro variable for use in subsequent steps as a foreign key value. The next step adds a record to the ValueLists dataset with the primary key OID="SCTESTVALS". The next step adds a record to the ItemValueListRefs dataset which associates the value list SCTESTVALS, specifically to the SCTEST column OID. The next step adds two records (one for Height and one for Weight) to the ItemDefs dataset, with the detailed value level metadata. Finally, the last step adds two records to the ValueListItemRefs dataset to associate the item values to the SCTESTVALS value list.

Add the code shown in Figure 14 to the code editor of the User Written Transform. This code runs the appropriate Clinical Standards Toolkit macro to create the CRT-DDS define.xml file, using the SAS CRT-DDS tables updated in the previous steps.

PhUSE 2010

```
*reassign srcdata to location of _svWork datasets;
data _null_;
path=pathname('_svwork');
rc=libname('srcdata');
rc=libname('srcdata',path);
run;
*create new define.xml file using updated SAS CRT-DDS datasets;
%crtdds_write(_cstCreateDisplayStyleSheet=1);
```

Figure 14 – Sample code for creating define.xml using customized SAS CRT-DDS datasets

Save the changes to the User Written Transform, and run the job. Based on these updates to the SAS CRT-DDS datasets, Figure 15 displays the value level metadata in the define.xml file.

```
<ItemDef OID="COL58" Name="SCTEST" DataType="text" Length="40"
SASFieldName="SCTEST" def:Label="Subject Characteristic">
  <def:ValueListRef ValueListOID="SCTESTVALS"/>
</ItemDef>

<def:ValueListDef OID="SCTESTVALS">
  <ItemRef ItemOID="VAL001" OrderNumber="1"
Mandatory="Yes" KeySequence="1"/>
  <ItemRef ItemOID="VAL002" OrderNumber="2"
Mandatory="Yes" KeySequence="2"/>
</def:ValueListDef>

<ItemDef OID="COL8" Name="SCTEST" DataType="text" Length="40"
SASFieldName="SCTEST"
def:Label="Subject Characteristic">
  <def:ValueListRef ValueListOID="SCTESTVALS"/>
</ItemDef>

<ItemDef OID="VAL001" Name="Height" DataType="text" Length="3"
SASFieldName="HEIGHT" Comment="Height taken barefoot"
def:Label="Height in inches"/>
<ItemDef OID="VAL002" Name="Weight" DataType="text" Length="4"
SASFieldName="WEIGHT" Comment="Weight without shoes"
def:Label="Weight in pounds"/>
```

Figure 15 – Partial define.xml file showing ValueList metadata

Validating the Define.xml File

SAS Clinical Data Integration also includes a transformation which allows the user to validate the define.xml file against the CRT-DDS 1.0.0 XML schema. If you have customized the CRT-DDS file with additional metadata, it is wise to run the CRT-DDS Validation transform on the resulting CRT-DDS file. To use this transformation connect the define.xml file to be validated to the CRT-DDS Validation transformation as shown in Figure 16.

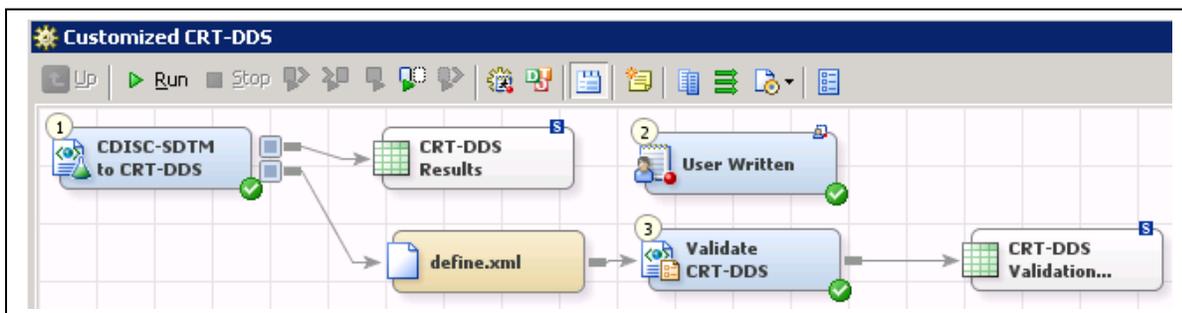


Figure 16 – Customized CRT-DDS Process Flow with CRT-DDS Validation

When the job is run, the results of the validation are written to the attached validation_results dataset. Information written to the validation_results dataset includes any error messages and the line and column number where the error was found. Figure 17 below shows an example of errors which might be found.

PhUSE 2010

12	(Line 20/Column 90) cvc-attribute.3: The value 'Y' of attribute 'Mandatory' on element 'ItemRef' is not valid with respect to its type, 'YesOrNo'. ...	Error
13	(Line 20/Column 90) cvc-enumeration-valid: Value 'Y' is not facet-valid with respect to enumeration '[Yes, No]'. It must be a value from the enume...	Error

Figure 17 –CRT-DDS Validation Results

CONCLUSION

The CRT-DDS file is a representation of the metadata collected during the study. The ability to centrally manage metadata facilitates the consistency of the resulting define.xml file. Collection of the metadata as the study is in progress reduces the time needed to create the define.xml file for the submission. In addition to providing central metadata management, SAS solutions provide for the transformation of the metadata into the CRT-DDS define.xml format.

ACKNOWLEDGMENTS

Jan Kratky and Pete Villiers, SAS Software Developers, for reviewing this paper and providing content related to CRT-DDS generation.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Julie Maddox	Mark Lambrecht
SAS Institute Inc.	SAS Institute Inc.
SAS Campus Drive	Hertenbergstraat 6
Cary, NC, 27519, USA	B-3080 Tervuren, Belgium

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.

APPENDIX A - CDISC-CRTDDS SAS REPRESENTATION

DefineDocument – contains information specifically about the define.xml file, including description and file creation date/time

Study – contains descriptive information about each study included in the define.xml file

MetaDataVersion – contains information about the data standards used in the study.

AnnotatedCRF – contains document references to the annotated case report form

SupplementalDocs - contains document references to the supplemental documentation

MDVLeaf – contains the href link for each of the documents listed in the AnnotatedCRF and SupplementalDocs tables.

MDVLeafTitles – contains a descriptive title for each MDVLeaf item.

Presentation – contains how information on the study is presented

ProtocolEventRefs - contains a list of study events and the order they occur

StudyEventDefs – contains descriptive information on the study events

FormDefs - contains a list of forms for the study

StudyEventFormRefs – associates a particular form to a specific study event

FormDefArchLayouts - contains document references for the form layouts

CodeLists – contains name, type and SAS Format name

ExternalCodeLists – contains name and versions of external code lists

CodeListItems – contains a list of coded items for each code list

CLItemDecodeTranslatedText – contains the full text for a coded item in the specified language

ImputationMethods – contains description of imputation methods

ComputationMethods – contains description of computation methods

ItemGroupDefItemRefs – associates the imputation methods to a specific itemdef (column)

ItemDefs – associates the computation method to a specific itemdef(column)

ValueLists – contains id of value lists

ValueListItemRefs– contains id of each item in a value list

ItemValueListRefs – associates each value list item to a row in the ItemDefs table

ItemGroupDefs – contains metadata for each of the tables included in the study

ItemGroupAliases – contains context and alias names for the table

ItemGroupLeaf - contains href location to the specified table

ItemGroupLeafTitles - contains descriptive title to table location

FormDefItemGroupRefs – associates a table with a form definition

ItemGroupDefItemRefs – associates the columns to the study tables

ItemDefs - contains the column metadata

CodeLists - contains name, type and SAS Format name

ItemRangeChecks - contains information on range checks for data in a column

ItemMURRefs – associates a column to a measurement unit

ItemQuestionTranslatedText - contains translated text of questions

ItemQuestionExternal – contains the name and version of external question dictionaries

ItemRole – contains the role of the column

ItemAliases – contains context and alias names for the column