Creating Define-XML version 2 including Analysis Results Metadata with the SAS® Clinical Standards Toolkit

Lex Jansen, SAS Institute Inc., Cary, NC, USA

ABSTRACT

Analysis Results Metadata provide traceability for a given analysis result to the specific Analysis Dataset Model (ADaM) data that were used as input to generating the analysis result; they also provide information about the analysis method used and the reason the analysis was performed. Analysis Results Metadata will assist the reviewer by identifying the critical analyses, providing links between results, documentation, and datasets, and documenting the analyses performed.

This paper describes the Analysis Results Metadata extension 1.0 for Define-XML version 2, and also, how the SAS® Clinical Standards Toolkit 1.7.1 can be used to create Define-XML version 2.0 documents including Analysis Results Metadata.

This paper assumes that the reader is familiar with some basic XML concepts, and also with Define-XML version 2. An earlier paper by the author (see references [1]) contains both a short overview of the XML needed to understand this paper, and also an overview of the structure of a define.xml file based on Define-XML version 1.0.0. A detailed overview of differences between CRT-DDS version 1.0.0 and Define-XML version 2 can be found in “Define-XML v2 - What’s New” [2].

Keywords: CDISC, Define-XML, define.xml, metadata, ADaM, Analysis Results Metadata, SAS Clinical Standards Toolkit

INTRODUCTION

In March 2013 the final version of the Define-XML 2.0.0 standard [3], formerly known as CRT-DDS (Case Report Tabulation Data Definition Specification) or “define.xml”, as most people called it, was released by the CDISC XML Technologies team. Define-XML 2.0.0 is a major revision of the Define-XML standard for transmission of SDTM, SEND and ADaM metadata. Version 1.0.0 was released for implementation in February 2005 [4]. Define-XML has been a useful mechanism and critical component for providing Case Report Tabulations Data Definitions in an XML format for CDISC based electronic submissions to a regulatory authority such as the U.S. Food and Drug Administration (FDA). In August 2013 the FDA started accepting Define-XML version 2.0.0 [5]. In the latest version of Study Data Technical Conformance Guide (version 3.3, March 2017) the FDA states that Define-XML version 2.0.0 is the preferred version of the Define-XML format [6]. In March 2016 the FDA had already announced the Support end date for CRT-DDS version 1.0.0 [7].

In January 2015 CDISC published the Analysis Results Metadata (ARM) as an extension to the Define-XML 2.0.0 standard for the purpose of submissions to regulatory agencies such as the FDA as well as for the exchange of analysis datasets and key results between other parties.

Sponsors are now able to submit analysis results metadata as part of their submission to a regulatory agency like the FDA in the USA or the PMDA in Japan. Although the FDA has not yet referenced analysis results metadata as a required part of an electronic submission as of the writing of this paper (March 2017), they have mentioned in a presentation that there are sponsors who want to submit analysis results metadata as part of a submission [8].

The Japanese Pharmaceuticals and Medical Devices Agency (PMDA) have mentioned in their Technical Conformance Guide on Electronic Study Data Submissions (2015, [9]):

> "In order for the review of clinical study data to progress smoothly, it is important that the relationship between the analysis results shown in the application documents and the analysis datasets is easily understandable. Therefore, the definition documents of the ADaM datasets should preferably include

> Analysis Results Metadata, which shows the relationship between the analysis results and the corresponding analysis dataset and the variables used, for the analyses performed to obtain the main results of efficacy and safety and clinical study results that provide the rationales for setting of the dosage and administration".

The PMDA mentions further in the same document:

> "For the format of the Analysis Results Metadata, the applicant should refer to the Analysis Results Metadata specification for Define-XML by CDISC to the extent possible, but if it is difficult to include it into the definition document, it is possible to submit it as a separated file in PDF format".

Even in the case that sponsors are not required to submit analysis results metadata in an electronic submission, it adds significant value to a submission. A referenced paper by DiIorio and Abolafia (2016) gives good reasons why results level metadata can also provide value to sponsors and their agents [10].

BACKGROUND

When submitting clinical study data in electronic format to the FDA, or other regulatory agencies, not only information from trials has to be submitted, but also information to help understand the data. Part of this information is a data definition file, which is the metadata describing the format and content of the submitted electronic data sets. When submitting data in CDISC format it is required to submit the data definition file in the Define-XML specification format (define.xml) as prepared by the CDISC define.xml team.

A Define-XML file is a structured data definition specification in a machine-readable XML format that provides different kinds of metadata for:

- The Study:
  - Study name, study description, protocol name
- Datasets:
  - Name, domain, label, class, structure, purpose, keys, data location, comments, documentation
- Variables:
  - Name, label, type, length, controlled terminology, origin, significant digit, display format, derivations, comments, documentation
- Variables under a condition:
  - Value level metadata or parameter value level metadata with the same kind of metadata as for 'regular' variables
- Controlled Terminology:
  - Standard or sponsor defined, name, type, valid values, decodes, reference to NCI code, external terminologies
- Derivations or Algorithms
- Comments
- Links to submission files:
  - Annotated CRF, Reviewers' Guides, source code files

The Analysis Results Metadata extension for Define-XML standard provides the metadata needed for traceability from a result used in a statistical display to the data in the analysis datasets:

- Identifiers and titles for the analysis displays (tables, figures) in the clinical study report
- Description of the specific analysis result within a display
- Purpose and reasons for performing the analysis
- The analysis parameter that is the focus of the analysis result

- Variables subject to analysis
- Datasets used to generate the analysis result
- Selection criteria for the records subject to analysis
- Corresponding description in the statistical analysis plan, analysis program name, and summary of the analytical methods
- Extract of the analysis program corresponding to the analysis method

The concept of results metadata for ADaM analyses and fields for representing this metadata were introduced in 2006 by CDISC with the publication of the Analysis Data Model (ADaM) Version 2.1 [11]. The document mentions that it is a best practice to provide this metadata to assist the reviewer by identifying the critical analyses, providing links between results, documentation, and datasets, and documenting the analyses performed. Analysis results metadata are not needed or even advisable for every analysis included in a clinical study report or submission. The sponsor determines which analyses should have analysis results metadata.

The key components in ADaM Analysis Results Metadata are:

- Analysis Display metadata definitions
    - Analysis Result metadata definitions
        - Analysis parameter(s)
        - Analysis dataset(s)
            - Analysis variable(s)
            - Selection criteria
        - Documentation
        - Programming statements

The table below shows the table from the Analysis Data Model (ADaM) document with the metadata fields to be used to describe an analysis result. The word “Display” is used instead of “Table” as it is more generic, referring to tabular or graphical presentation of results.

Table 1 Analysis Results Metadata Fields (Analysis Data Model (ADaM) Version 2.1 document)

Analysis Results Metadata Field	Description
DISPLAY IDENTIFIER	A unique identifier for the specific analysis display (such as a table or figure number)
DISPLAY NAME	Title of display, including additional information if needed to describe and identify the display (e.g., analysis population)
RESULT IDENTIFIER	Identifies the specific analysis result within a display. For example, if there are multiple p-values on a display and the analysis results metadata specifically refers to one of them, this field identifies the p-value of interest. When combined with the display identifier provides a unique identification of a specific analysis result.
PARAM	The analysis parameter in the BDS analysis dataset that is the focus of the analysis result. Does not apply if the result is not based on a BDS analysis dataset.
PARAMCD	Corresponds to PARAM in the BDS analysis dataset. Does not apply if the result is not based on a BDS analysis dataset.
ANALYSIS VARIABLE	The analysis variable being analyzed

Analysis Results Metadata Field	Description
REASON	The rationale for performing this analysis. It indicates when the analysis was planned (e.g., "Pre-specified in Protocol," "Pre-specified in SAP," "Data Driven," "Requested by Regulatory Agency") and the purpose of the analysis within the body of evidence (e.g., "Primary Efficacy," "Key Secondary Efficacy," "Safety"). The terminology used is sponsor defined. An example of a reason is "Primary Efficacy Analysis as Pre-specified in Protocol."
DATASET	The name of the dataset used to generate the analysis result. In most cases, this is a single dataset. However, if multiple datasets are used, they are all listed here.
SELECTION CRITERIA	Specific and sufficient selection criteria for analysis subset and / or numerator – a complete list of the variables and their values used to identify the records selected for the analysis. Though the syntax is not ADaM-specified, the expectation is that the information could easily be included in a WHERE clause or something equivalent to ensure selecting the exact set of records appropriate for an analysis. This information is required if the analysis does not include every record in the analysis dataset.
DOCUMENTATION	Textual description of the analysis performed. This information could be a text description, pseudo code, or a link to another document such as the protocol or statistical analysis plan, or a link to an analysis generation program (i.e., a statistical software program used to generate the analysis result). The contents of the documentation metadata element contains depends on the level of detail required to describe the analysis itself, whether or not the sponsor is providing a corresponding analysis generation program, and sponsor-specific requirements and standards. This documentation metadata element will remain free form, meaning it will not become subject to a rigid structure or controlled terminology.
PROGRAMMING STATEMENTS	The software programming code used to perform the specific analysis. This includes, for example, the model statement (using the specific variable names) and all technical specifications needed for reproducing the analysis (e.g., covariance structure). The name and version of the applicable software package should be specified either as part of this metadata element or in another document, such as a Reviewer's Guide (see Appendix B for more information about a Reviewer's Guide).

It must be recognized that analysis results metadata fields in the Analysis Data Model (ADaM) Version 2.1 document are described in less technical terms and without implementation details.

Before there was an official specification for submitting analysis results metadata there were several CDISC Pilot Projects where the usefulness of the concept was demonstrated [12]. Since there was no formal specification for the submission vehicle of analysis results metadata, the pilots used their own informal specifications.

The Analysis Results Metadata extension to the Define-XML version 2 specification document ([13]) handles the technical implementation of what is described in Analysis Data Model (ADaM) Version 2.1 document.
Along with the Analysis Results Metadata v1.0 specification, the distribution package includes:

- The Analysis Results Metadata 1.0 XML schema
- An ADaM based Define-XML version 2.0 example and its HTML rendition
- A sample XSL stylesheet for the Define-XML example

THE FOUNDATION FOR THE ANALYSIS RESULTS METADATA EXTENSION

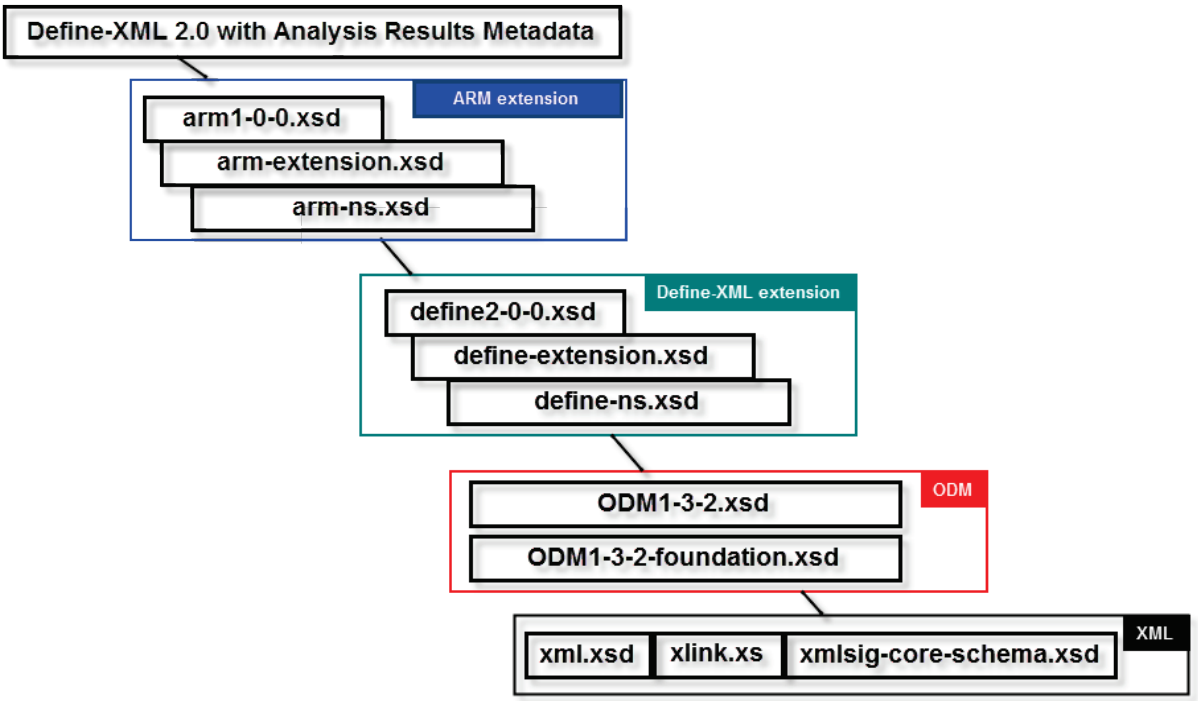
The Analysis Results Metadata for Define-XML standard is based on the CDISC Operational Data Model (ODM) XML schema [14]. The new Define-XML version 2 takes full advantage of the latest ODM version 1.3.2.

The CDISC Operational Data Model is a vendor neutral, platform independent XML format for interchange and archival of clinical study data. The model represents study metadata, administrative metadata, reference data and subject data associated with a clinical trial. The ODM format is defined by an XML schema and a specification.

One of the features of the ODM is a standardized mechanism for defining schema extensions to provide functionality needed to support interchange requirements for specialized use cases.

To address the specific needs of data transmission in support of regulatory submissions, CDISC has developed the Define-XML model, which is implemented as an extension to the ODM foundation schema. These extensions follow the guidelines for Vendor Extensions provided in the ODM specification and comply with the W3C XML Schema 1.0 specification. The XML schema files for the Define-XML standard are available online [3]. Figure 1 depicts the extension mechanism.

Figure 1 Analysis Results Metadata 1.0 for Define-XML v2 as an ODM 1.3.2 extension



Although Define-XML version 1 has been a successful standard since 2005, there were good reasons to use Define-XML version 2.0 for the Analysis Results Metadata extension.

ARM uses several features that were new in the Define-XML version 2 standard:

- Explicit support for document references.
- Referencing short comments self-contained in the Define-XML document or long comments referenced in external documents
- The Where Clause mechanism is used for implementing selection criteria definitions that describe analysis data subsets

Figure 2 shows how the Analysis Results Metadata extension fits in a Define-XML version 2.0 document. Apart from the inclusion of the Analysis Results Metadata (arm:AnalysisResultDisplays) element, it is needed to add content to 3 more sections:

- Where Clause Definitions (def:WhereClauseDef)
- Comments (def:CommentDef)
- Referenced Documents (def:leaf)

Figure 2 Structure of Define-XML v2 with Analysis Results Metadata 1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:def="http://www.cdisc.org/ns/def/v2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:arm="http://www.cdisc.org/ns/arm/v1.0"
  ODMVersion="1.3.2" FileType="Snapshot" FileOID="CDISC-Sample"
  CreationDateTime="2014-03-28T11:07:23:00"
  Originator="CDISC ADaM Metadata Team">
  <Study OID="cdisc01">
    <GlobalVariables>
      <StudyName>CDISC Sample</StudyName>
      <StudyDescription>CDISC-Sample Data Definitions</StudyDescription>
      <ProtocolName>CDISC-Sample</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="MDV.CDISC01.ADaMIG.1.0.ADaM.2.1"
      Name="Study CDISC-Sample, Data Definitions"
      Description="Study CDISC01, Data Definitions"
      def:DefineVersion="2.0.0"
      def:StandardName="ADaM-IG"
      def:StandardVersion="1.0">
```

< Supplemental Data Definitions (def:SupplementalDoc) >

< Value Level Metadata (def:ValueListDef) >

< Where Clause Definitions (def:WhereClauseDef) >

< Domain Level Metadata (ItemGroupDef) >

< Variable Level Metadata (ItemDef) >

< Controlled Terminology Metadata (CodeList) >

< Computational Algorithms (MethodDef) >

< Comments (def:CommentDef) >

< Referenced Documents (def:leaf) >

< Analysis Results Metadata (arm:AnalysisResultDisplays) >

```
    </MetaDataVersion>
  </Study>
</ODM>
```

SAS® CLINICAL STANDARDS TOOLKIT

The SAS Clinical Standards Toolkit focuses on standards defined by the Clinical Data Interchange Standards Consortium (CDISC). CDISC is a global, open, multidisciplinary, nonprofit organization that has established standards to support the acquisition, exchange, submission, and archival of clinical research data and metadata. The CDISC mission is to develop and support global, platform-independent data standards that enable information-system interoperability, which, in turn, improves medical research and related areas of health care. The SAS Clinical Standards Toolkit is not limited to supporting CDISC standards. The SAS Clinical Standards Toolkit framework is designed to support the specification and use of any user-defined standard.

SAS Clinical Standards Toolkit 1.6 and newer already had the capability to create a complete Define-XML version 2.0 document. With SAS Clinical Standards Toolkit 1.7.1 a Define-XML version 2.0 document can now also include the Analysis Results Metadata version 1.0 extension, as published by CDISC. An earlier paper by this author showed how to create a Define-XML version 2.0 document with the SAS Clinical Standards Toolkit 1.6 [15].

The SAS Clinical Standards Toolkit 1.7 (see reference [16]) includes support for the following CDISC standards:

- SDTM 1.3.1, 3.1.2, 3.1.3 and 3.2
- An initial implementation of the CDISC SEND 3.0 standard, including definition of all domains and columns.
- ADaM 2.1 (ADSL, Basic Data Structure, ADAE) and Analysis Results Metadata templates, as well as new validation checks in support of ADAE and ADTTE.
- CRT-DDS 1.0 (define.xml), including define.pdf and Value Level metadata support.
- Dataset-XML 1.0:
    - Creating Dataset-XML files from SAS data sets
    - Creating SAS data sets from Dataset-XML files
    - Validating Dataset-XML files against an XML schema
    - Comparing original SAS data sets with SAS data sets created from Dataset-XML files
- Define-XML 2.0:
    - A complete definition of the metadata model for CDISC Define-XML 2.0 (including Analysis Results Metadata after applying a hot fix)
    - Creation of a complete Define-XML 2.0 file based on study metadata, with study metadata examples from SDTM 3.2 and ADaM 2.1
    - Validation of a Define-XML 2.0 file against the XML schema definition, as published by CDISC
    - Import of a Define-XML 2.0 file into the SAS representation of the Define-XML 2.0 metadata model
    - support of creating an initial version of the SAS source metadata data sets (source_study, source_tables, source_columns, source_codelists, source_values, source_documents, and source_analysisresults) that serve as input for creating a Define-XML v2.0 file
- ODM 1.3.0 and ODM 1.3.1, including support for the extraction of ODM Clinical data and ODM Reference data into SAS data sets.
- The implementation of CT 1.0.0, a tool to support the import of NCI CDISC Controlled Terminology in the ODM XML format into SAS data sets and SAS format catalogs.
- CDISC Controlled Terminology packages that includes terminology sets as posted to the NCI FTP site.
- Furthermore, a set of macro tools to validate the SAS Clinical Standards Toolkit metadata itself ("Internal validation").
- Metadata Management tools to add, update and delete Toolkit metadata

SAS Clinical Standards Toolkit 1.7 Second Edition (also known as SAS Clinical Standards Toolkit 1.7.1) is a hot fix that adds support for the CDISC Analysis Results Metadata specification version 1.0 for Define-XML version 2.0, and

also adds the December 2015 snapshot of CDISC NCI controlled terminology for ADaM. SAS Clinical Standards Toolkit 1.7.1 is supported with SAS 9.4 (TS1M3) or later on Windows x64 and Linux x64.

Each SAS Clinical Standards Toolkit standard provides a SAS representation of the published source guidelines or source specification. The SAS representation is designed to serve as a model or template of the source specification. Two key design requirements shaped the implementation of the SAS Clinical Standards Toolkit standards.

1) Each supported standard is represented in one or more SAS files. This facilitates these points:
   - It provides SAS users with an implementation of data models and standards that are based on SAS.
   - It enables you to use SAS routines to assess how well any user-defined set of data and metadata conforms to the standard.
   - It enables you to use SAS code to read and derive files in other formats (for example, XML).

   Each SAS Clinical Standards Toolkit standard is an optimized reference standard from a SAS perspective.
2) You are able to define your own customized standards, or you are able to modify existing SAS standards.

Since a Define-XML file does not have a 2-dimensional data structure, it is not a trivial task to translate this hierarchical file to a number of SAS data set with rows and columns. SAS has defined a relational data model that represents a Define-XML file.

The highly structured nature of CDISC Define-XML data requires that any mapping to a relational format include a large number of data sets, with foreign key relationships to help preserve the intended non-relational object structure. In the SAS Clinical Standards Toolkit, foreign key relationships are enforced when validating the CDISC Define-XML data sets.

CREATING DEFINE-XML VERSION WITH THE SAS CLINICAL STANDARDS TOOLKIT

There are three key macros that are provided with the SAS Clinical Standards Toolkit that support creation of a CDISC Define-XML 2.0 define.xml file. The three macros are listed in the order in which they are executed:

1. The **define_sourcetodefine** macro creates the tables for the SAS representation of the Define-XML 2.0 files from study metadata. This macro, using SDTM or ADaM table and column metadata as its source, populates a subset of the Define-XML 2.0 data sets.
2. The **define_write** macro creates the define.xml file from the SAS representation of the CDISC Define-XML 2.0 files.
3. The **cstutilxmlvalidate** macro validates that the XML file is syntactically correct according the XML schema that is associated with the CDISC Define-XML 2.0 standard.

These macros are called by driver programs that are responsible for properly setting up each SAS Clinical Standards Toolkit process to perform a specific SAS Clinical Standards Toolkit task. Several sample driver programs are provided with the SAS Clinical Standards Toolkit CDISC Define-XML 2.0 standard related to the creation of the define.xml file.

Here is the purpose of each of these driver programs:

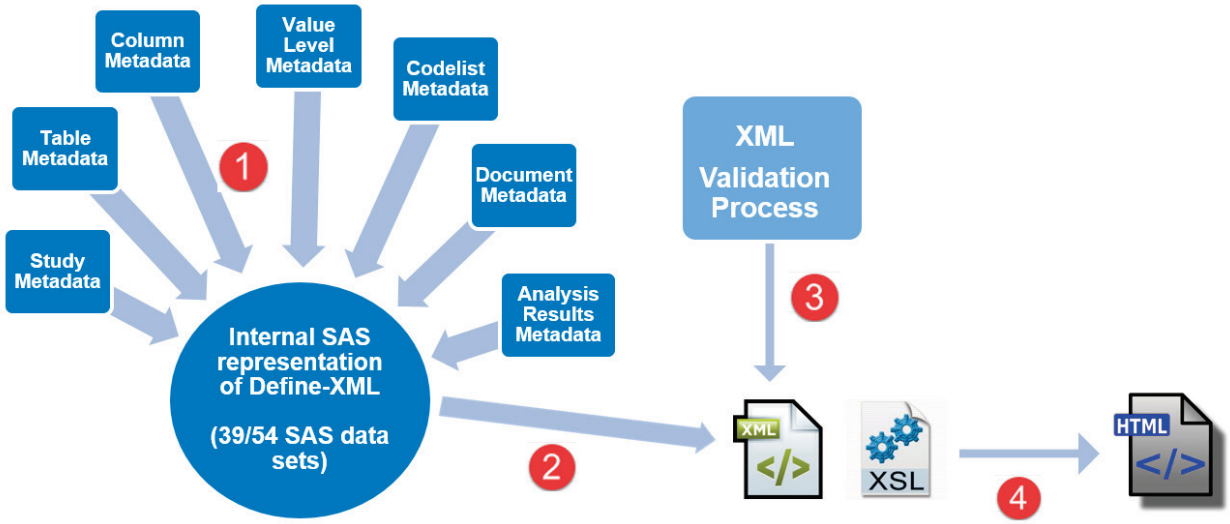
- The create_sasdefine_from_source.sas driver program sets up the required metadata and SASReferences data set for the sample study. It runs the **define_sourcetodefine** macro. It creates the SAS representation of the CDISC Define-XML 2.0 data sets from the sample study source metadata data sets.
- The create_definexml.sas driver program creates the CDISC Define-XML 2.0 define.xml file from the SAS representation of the CDISC Define-XML 2.0 data sets. It runs the **define_write** and **cstutilxmlvalidate** macros. This driver program creates and validates the XML syntax for the define.xml file.
- The create_definexml_from_source_adam.sas combines the previous two driver programs into one driver program.

These driver programs are examples that are provided with the SAS Clinical Standards Toolkit. You can use these driver programs or create your own. The names of these driver programs are not important. However, the content is important and demonstrates how the various SAS Clinical Standards Toolkit framework macros are used to generate the required metadata files.

Figure 3 shows the complete process for creating a complete Define-XML version 2 in the SAS Clinical Standards Toolkit. The following steps are identified:

1. Run the **define_sourcetodefine** macro to create the tables for the SAS representation of the Define-XML 2.0 files from study metadata.
2. Run the **define_write** macro to create the define.xml file from the SAS representation of the CDISC Define-XML 2.0 files.
3. Run the **cstutilxmlvalidate** macro to validate the define.xml file against the XML schema that is associated with the CDISC Define-XML 2.0 standard.
4. Run PROC XSL to create an HTML document from the define.xml file and the XSL stylesheet that comes with the SAS Clinical Standards Toolkit.

Figure 3 The SAS macro process to create a valid Define-XML 2.0 document



For Define-XML 2.0 the following source metadata SAS data sets are defined in SAS Clinical Standards Toolkit 1.7.1:

- source_study:
  - Metadata about the study, such as study name, study description and protocol name.
- source_tables:
  - Domain metadata, such as name, domain, description (label), class, structure, purpose, keys, data location, comments and documentation reference.
- source_columns
  - Column metadata, such as name, description (label), order number, datatype, length, codelist, origin, significant digits, display format, derivation (algorithm), comments and documentation reference.
- source_values
  - Value level metadata, where a condition is defined in the WHERECLAUSE column.
    Example WHERECLAUSE values are:
    - (LBTESTCD EQ "BILI") AND (LBCAT EQ "CHEMISTRY") AND (LBSPEC EQ "BLOOD")
    - VSTESTCD EQ “HEIGHT”
    - PARAMCD IN ("ACITM01","ACITM02","ACITM03")
    - PARAMCD NOTIN ("ACTOT")

    The column which the value level metadata is attached to, is defined by the TABLE and COLUMN

columns. Apart from the WHERECLAUSE column, this data set contains the same kind of metadata as the source_columns data set.

- source_codelists:
    - Metadata related to Controlled Terminology, such as name, description, datatype, SAS formatname, valid values, decodes, rank, order number, reference to NCI code, external terminologies
- source_documents:
    - Metadata related to referenced documents, such as annotated CRF, reviewer guides or other supplemental documents. Records in this data set can be linked to source_tables, source_columns, source_values, or source_analysisresults data sets by the combination of the TABLE, COLUMN, WHERECLAUSE, DISPLAYIDENTIFIER and RESULTIDENTIFIER columns. Page numbers and named destinations in PDF files can be defined in this data set as well. Documents are attached to comments, methods or origins based on the value of the DOCTYPE column.
      Valid values of the DOCTYPE column in SAS Clinical Standards Toolkit 1.7 were: COMMENT, CRF or METHOD. Version 1.7.1 adds the following values: SUPPDOC, DISPLAY, RESULTDOC and RESULTCODE.
- Source_analysisresults:
    - Metadata related to analysis displays and results: display identifier, display name, display description, result identifier, result description, analysis purpose and reason, parameter column, analysis variables, analysis datasets, selection criteria (WhereClause), Selection criteria for the records subject to analysis, result programming code and context, result documentation.

The complete structure of the SAS source metadata data sets can be found in Appendix 1.

The **define_sourcetodefine** macro extracts data from the SDTM or ADaM study source metadata files. Depending on the available source information, the macro attempts to convert the information into the tables that represent the SAS interpretation of the CDISC Define-XML 2.0 model.

The parameters in table 2 must be set before submitting the **define_sourcetodefine** macro.

Table 2 define_sourcetodefine macro parameters

Parameter	Required	Description
_cstOutLib	Yes	The library reference where the resulting tables are written.
_cstSourceTables	Yes	The data set that contains the metadata for the domains to include in the Define-XML file.
_cstSourceStudy	Yes	The data set that contains the metadata for the studies to include in the Define-XML file.
_cstSourceColumns	Yes	The data set that contains the metadata for the Domain columns to include in the Define-XML file.
_cstSourceCodeLists	No	The data set that contains the metadata for the CodeLists to include in the Define-XML file.
_cstSourceValues	No	The data set that contains the metadata for the Value Level columns to include in the Define-XML file.
_cstSourceDocuments	No	The data set that contains the metadata for document references to include in the Define-XML file.
_cstSourceAnalysisResults	No	The data set that contains the metadata for analysis results to include in the Define-XML file.
_cstFullModel	Yes	Create all data sets in the Define-XML model (Y) or only the core model (N). The default is N.
_cstLang	No	The ODM TranslatedText/@lang attribute.
_cstCheckLengths	Yes	Check the actual value lengths of character variables against the lengths defined in the metadata templates.

Here is an example of a call to the define_sourcetodefine macro:

```
%define_sourcetodefine(
  _cstOutLib=srcdata,
  _cstSourceStudy=sampdata.source_study,
  _cstSourceTables=sampdata.source_tables,
  _cstSourceColumns=sampdata.source_columns,
  _cstSourceCodeLists=sampdata.source_codelists,
  _cstSourceDocuments=sampdata.source_documents,
  _cstSourceValues=sampdata.source_values,
  _cstSourceAnalysisResults=sampdata.source_analysisresults,
  _cstFullModel=N,
  _cstCheckLengths=Y,
  _cstLang=en
  );
```

In the example, the define_sourcetodefine macro extracts data from the SDTM, SEND, or ADaM source metadata tables in the sampdata library and writes all of the tables that represent the SAS interpretation of the CDISC Define-XML 2.0 model to the SAS srcdata library.

A source study data set (source_study) can only have one record and is required by this macro. The variables in Table 3 are all required in this data set.

Table 3 Variables required in the Source Study Metadata Data Set (source_study)

Variable	Description
SASRef	Libname reference.
StudyName	The name of the study. This value is used to populate the srcdata.study.studyname column.
StudyDescription	The description of the study. This value is used to populate the srcdata.study.studydescription column. Note: You cannot use commas, semicolons, or quotation marks in the description.
ProtocolName	The name of the protocol for the study. This value is used to populate the srcdata.study.protocolname column.
StudyVersion	The name of the define document to create. This value is used to populate the srcdata.metadataversion.oid column.
FormalStandardVersion	The formal version of the standard as used in Define-XML 2.0 This value is used to populate the srcdata.definedocument.standardversion column (For example, “1.0” for ADaM.)
FormalStandardName	The formal name of the standard as defined by the Define-XML 2.0 specification. This value is used to populate the srcdata.definedocument.standardname column (For example, ADaM-IG.)
Standard	The name of the standard as it is registered in the SAS Clinical Standards Toolkit. (For example, CDISC-ADAM.)
StandardVersion	The version of the standard as it is registered in the SAS Clinical Standards Toolkit. (For example, 2.1.)

Only a single study can be referenced in the source data sets. The define_sourcetodefine macro will only select records from the source_tables, source_colums, source_codelists, source_values, source_documents, and source_analysisresults data sets whose STUDYVERSION column value is equal to the value of the STUDYVERSION column in the source_study data set. Figure 4 shows an example of the source_study data set.

Figure 4 Example of the source_study data set.

VIEWTABLE: Sampdata.Source_study (Source Study Metadata)

	sasref	studyname	studydescription	protocolname	studyversion
1	SRCDATA	CDISC-Sample	CDISC-Sample Data Definition	CDISC-Sample	MDV.CDISC01.ADaMIG.1.0.ADaM.2.1

VIEWTABLE: Sampdata.Source_study (Source Study Metadata)

	formalstandardname	formalstandardversion	standard	standardversion
1	ADaM-IG	1.0	CDISC-ADAM	2.1

Figures 5 and 6 show examples of the source_tables and source_columns data sets. Not all columns are displayed in these examples.

Figure 5 Example of the source_tables data set (partial).

VIEWTABLE: Sampdata.Source_tables (Source Table Metadata)

	table	label	order	domain	class	xmlpath	xmltitle	structure	purpose	keys	comment
1	ADAE	Adverse Events Analysis	3		OCCURRENCE DATA STRUCTURE	../transport/cdisc-adam-2.1/adae.xpt	adae.xpt	one record per subject per adverse event	Analysis	STUDYID USUBJID AETERM ASTDT AESEQ	See SAS program
2	ADQSADAS	ADAS-Cog Analysis	2		BASIC DATA STRUCTURE	../transport/cdisc-adam-2.1/adqsadas.xpt	adqsadas.xpt	One record per subject per parameter per analysis visit per analysis date	Analysis	STUDYID USUBJID PARAMCD AVISIT ADT	See referenced dataset creation program and Analysis Data Reviewer's Guide, Section 2.1
3	ADSL	Subject Level Analysis	1		SUBJECT LEVEL ANALYSIS DATASET	../transport/cdisc-adam-2.1/adsl.xpt	adsl.xpt	one record per subject	Analysis	STUDYID USUBJID	Screen Failures are excluded since they are not needed for this study analysis. See Analysis Data Reviewer's Guide, page 6.

Figure 6 Example of the source_columns data set (partial).

table	column	label	order	length	displayformat	xmldatatype	xmlcodelist	origin	origindescription	algorithm	algorithmtype	formalexpression	formalexpressioncontext	comment
ADQSADA	STUDYID	Study Identifier	1	12		text		Predecessor	ADSL.STUDYID					
ADQSADA	SITEID	Study Site Identifier	2	3		text		Predecessor	ADSL.SITEID					
ADQSADA	SITEGR1	Pooled Site Group 1	3	3		text		Predecessor	ADSL.SITEGR1					
ADQSADA	USUBJID	Unique Subject Identifier	4	11		text		Predecessor	ADSL.USUBJID					
ADQSADA	TRTSDT	Date of First Exposure to Treatment	5	8	date9.	integer		Predecessor	ADSL.TRTSDT					
ADQSADA	TRTEDT	Date of Last Exposure to Treatment	6	8	date9.	integer		Predecessor	ADSL.TRTEDT					
ADQSADA	TRTP	Planned Treatment	7	20		text	CL.ARM	Predecessor	ADSL.TRT01P					
ADQSADA	TRTPN	Planned Treatment (N)	8	8		integer	CL.ARMN	Predecessor	ADSL.TRT01PN					
ADQSADA	AGE	Age	9	8		integer		Predecessor	ADSL.AGE					
ADQSADA	AGEGR1	Pooled Age Group 1	10	5		text	CL.AGEGR1	Predecessor	ADSL.AGEGR1					
ADQSADA	AGEGR1N	Pooled Age Group 1 (N)	11	8		integer	CL.AGEGR1N	Predecessor	ADSL.AGEGR1N					
ADQSADA	RACE	Race	12	32		text	CL.RACE	Predecessor	ADSL.RACE					
ADQSADA	RACEN	Race (N)	13	8		integer	CL.RACEN	Predecessor	ADSL.RACEN					
ADQSADA	SEX	Sex	14	1		text	CL.SEX	Predecessor	ADSL.SEX					
ADQSADA	ITTFL	Intent-To-Treat Population Flag	15	1		text	CL.YN	Predecessor	ADSL.ITTFL					
ADQSADA	EFFFL	Efficacy Population Flag	16	1		text	CL.YN	Predecessor	ADSL.FASFL					
ADQSADA	COMP24FL	Completers of Week 24 Population Flag	17	1		text	CL.YN	Predecessor	ADSL.COMP24FL					
ADQSADA	AVISIT	Analysis Visit	18	16		text	CL.AVISIT	Derived		Derived based on windowing algorithm described in SAP, Section 8.2	Computation			
ADQSADA	AVISITN	Analysis Visit (N)	19	8		integer	CL.AVISITN	Assigned						Numeric code for AVISIT

The example of the source_values data set in Figure 7, used to define Parameter Value Level Metadata, shows that this data set looks very much like the source_columns data set. It has an extra column to describe the condition (WHERECLAUSE) for which the metadata is defined. The TABLE and COLUMN variables tell which variable the value list is attached to.

The algorithmtype, formalexpression, and formalexpressioncontext variables were not part of the source_columns and source_values data set.

SAS Technical Support Usage Note 60134 [17] shows how to add the following variables to the source metadata data sets source_columns and source_values:

- algorithmtype char(11) label='Type of Algorithm',
- formalexpression char(1000) label='Formal Expression for Algorithm'
- formalexpressioncontext char(1000) label='Context to be used when evaluating the FormalExpression content'

After applying Technical Support Usage Note 60134 the customization of this metadata for Define-XML v2.0 is supported in SAS Clinical Standards Toolkit 1.7.1.

Figure 7 Example of the source_values data set (partial).

table	column	whereclause	label	length	xmldatatype	xmlcodelist	core	origin	origindescription	algorithm	algorithmtype	comment
ADQSADAS	AVAL	PARAMCD IN ("ACITM14", "ACITM10", "ACITM12", "ACITM04", "ACITM03", "ACITM02", "ACITM07", "ACITM08", "ACITM09", "ACITM05", "ACITM06", "ACITM11", "ACITM01", "ACITM13")	Analysis Value	8	integer			Derived		QS.QSSTRESN where QSTESTCD=PARAMC	Computation	
ADQSADAS	AVAL	PARAMCD EQ "ACTOT"	Analysis Value	8	integer			Derived		Sum of ADAS scores for items 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, and 14, see Analysis Data Reviewers Guide (Page 3) for details on adjusting for missing values.	Computation	
ADQSADAS	DTYPE	PARAMCD NE "ACTOT"	Derivation Type	7	text	CL.DTYPE		Assigned				Value: null
ADQSADAS	DTYPE	PARAMCD EQ "ACTOT"	Derivation Type	7	text	CL.DTYPE		Assigned				Value: LOCF denotes that the LOCF imputation method was used to impute the value for the given parameter and analysis visit.
ADQSADAS	QSSEQ	PARAMCD NE "ACTOT"	Sequence Number	8	integer			Predecessor	QS.QSSEQ			
ADQSADAS	QSSEQ	PARAMCD EQ "ACTOT"	Sequence Number	8	integer			Assigned				Set QSSEQ to missing for post baseline records. Set to QS.QSSEQ where QS.VISIT=BASELINE and QS.QSTESTCD=ACTOT.

Figure 8 shows an example of Controlled Terminology metadata. The source_codelists table is linked to the source_columns table and the source_values table through the SASFORMATNAME column:

source_codelists.CODELIST = source_columns.XMLCODELIST

source_codelists.CODELIST = source_values.XMLCODELIST

Figure 8 Example of the source_codelists data set (partial).

VIEWTABLE: Sampdata.Source_codelists (Source Codelist Metadata)

	codelist	codelistname	codelistncicode	codelistdatatype	sasformatname	codedvaluechar	codedvaluenum	decodetext	codedvaluencicode	rank	ordernumber	dictionary	version
1	CL.AECAUS	Causality		text		NONE	.			1	.		
2	CL.AECAUS	Causality		text		POSSIBLE	.			2	.		
3	CL.AECAUS	Causality		text		PROBABLE	.			3	.		
4	CL.AECAUS	Causality		text		REMOTE	.			4	.		
5	CL.AEDICT	Adverse Event Dictionary		text			.			.	.	MedDRA	8.0
6	CL.AESEV	Severity/Intensity Scale for Adverse Events	C66769	text	$AESEV	MILD	.	Grade 1	C41338	1	.		
7	CL.AESEV	Severity/Intensity Scale for Adverse Events	C66769	text	$AESEV	MODERATE	.	Grade 2	C41339	2	.		
8	CL.AESEV	Severity/Intensity Scale for Adverse Events	C66769	text	$AESEV	SEVERE	.	Grade 3	C41340	3	.		
9	CL.AGEGR1	Age Group		text		<65	.			1	.		
10	CL.AGEGR1	Age Group		text		65-80	.			2	.		
11	CL.AGEGR1	Age Group		text		>80	.			3	.		
12	CL.AGEGR1N	Age Group (N)		integer	AGEGR1N		1	<65		.	1		
13	CL.AGEGR1N	Age Group (N)		integer	AGEGR1N		2	65-80		.	2		
14	CL.AGEGR1N	Age Group (N)		integer	AGEGR1N		3	>80		.	3		
15	CL.AGEU	Age Unit	C66781	text		YEARS	.		C29848	.	.		
16	CL.ARM	Actual Treatment		text		Placebo	.			.	1		
17	CL.ARM	Actual Treatment		text		Xanomeline Low Dose	.			.	2		
18	CL.ARM	Actual Treatment		text		Xanomeline High Dose	.			.	3		

Figure 9 shows an example of the source_documents data set. This data set defines metadata for references to supporting documents like the annotated Case Report Form (SDTM), and other types of supporting documents, like the Statistical Analysis Plan, the Clinical Study Report, or the Analysis Data Reviewer's Guide [18]. This data set can also have metadata for references to files with SAS programming code.

Figure 9 Example of the source_documents data set (partial)

VIEWTABLE: Sampdata.Source_documents (Source Document Metadata)

	doctype	href	title	pdfpagereftype	pdfpagerefs	table	column	whereclause	displayidentifier	resultidentifier
1	COMMENT	../suppdocs/adae-sas.txt	adae.sas			ADAE				
2	COMMENT	../suppdocs/adqsadas-sas.txt	adqsadas.sas			ADQSADAS				
3	COMMENT	../suppdocs/analysis-data-reviewers-guide.pdf	Analysis Data Reviewer's Guide	NamedDestination	Section2.1	ADQSADAS				
4	COMMENT	../suppdocs/analysis-data-reviewers-guide.pdf	Analysis Data Reviewer's Guide	PhysicalRef	6	ADSL				
5	DISPLAY	../suppdocs/dummy-csr.pdf	Table 14-3.01	PhysicalRef	2				RD.Table_14-3.01	
6	DISPLAY	../suppdocs/dummy-csr.pdf	Table 14-5.02	PhysicalRef	3				RD.Table_14-5.02	
7	METHOD	../suppdocs/analysis-data-reviewers-guide.pdf	Analysis Data Reviewer's Guide	PhysicalRef	3	ADQSADAS	AVAL	PARAMCD EQ "ACTOT"		
8	RESULTCODE	../suppdocs/at14-5-02-sas.txt	at14-5-02.sas						RD.Table_14-5.02	AR.Table_14-5.02.R.1
9	RESULTDOC	../suppdocs/dummy-sap.pdf	Statistical Analysis Plan	PhysicalRef	4				RD.Table_14-3.01	AR.Table_14-3.01.R.2
10	RESULTDOC	../suppdocs/dummy-sap.pdf	Statistical Analysis Plan	PhysicalRef	5				RD.Table_14-5.02	AR.Table_14-5.02.R.1
11	RESULTDOC	../suppdocs/dummy-sap.pdf	Statistical Analysis Plan	PhysicalRef	4				RD.Table_14-3.01	AR.Table_14-3.01.R.1
12	SUPPDOC	../suppdocs/analysis-data-reviewers-guide.pdf	Analysis Data Reviewer's Guide							
13	SUPPDOC	../suppdocs/dummy-csr.pdf	Clinical Study Report							
14	SUPPDOC	../suppdocs/dummy-sap.pdf	Statistical Analysis Plan							

An explanation of the metadata in Figure 9 follows below.

Record 1-2 References to SAS programs attached to a comment in tables ADAE and ADQSADAS. The value "COMMENT" in the DOCTYPE column tells that these references are attached to a comment.

Record 3 A reference to the Analysis Data Reviewer's Guide (analysis-data-reviewers-guide.pdf) attached to a comment in the ADQSADAS table with a PDF NamedDestination specified (“Section2.1”). This NamedDestination has to be defined in the PDF document. The value "COMMENT" in the DOCTYPE column tells that these references are attached to a comment.

Record 4 A reference to page 6 in the Analysis Data Reviewer's Guide (analysis-data-reviewers-guide.pdf) attached to a comment in the ADQSADAS table. The value "COMMENT" in the DOCTYPE column tells that these references are attached to a comment.

Records 5-6 Table references from analysis displays to physical pages in the Clinical Study Report. The value "DISPLAY" in the DOCTYPE column tells that these references are attached to the display as a whole.

Record 7 A reference to page 3 in the Analysis Data Reviewer's Guide (analysis-data-reviewers-guide.pdf) to further explain a derivation (method) attached to an item defined by a WHERECLAUSE (Value Level Metadata), ADQSADA.AVAL where PARAMCD equals "ACTOT". The value "METHOD" in the DOCTYPE column tells that these references are attached to a method (algorithm).

Record 8 A reference from an analysis result within a display to a file with programming statements (at14-5-02.sas). The analysis result is identified by the DISPLAYIDENTIFIER and RESULTIDENTIFIER columns. The value "RESULTCODE" in the DOCTYPE column tells that these references are results documentation.

Records 9-11 A reference from an analysis result within a display to a physical page within the Statistical Analysis Plan (at14-5-02.sas). The analysis result is identified by the DISPLAYIDENTIFIER and RESULTIDENTIFIER columns. The value "RESULTDOC" in the DOCTYPE column tells that these references are results documentation.

Records 12-14 References to various documents: Analysis Data Reviewer's Guide, Clinical Study Report and Statistical Analysis Plan. No physical page numbers or named destination is defined. The value "SUPDDOC" in the DOCTYPE column tells that these references are not attached to a specific table, column, value level metadata item, display, or result.
These references will be included in the def:SupplementalDoc element in the Define-XML document. A stylesheet typically will display links to these documents in the bookmarks section of the HTML rendition of the Define-XML document, as can be seen in Figure 10.

Figure 10 Example of a rendition of a Define-XML document that shows document references

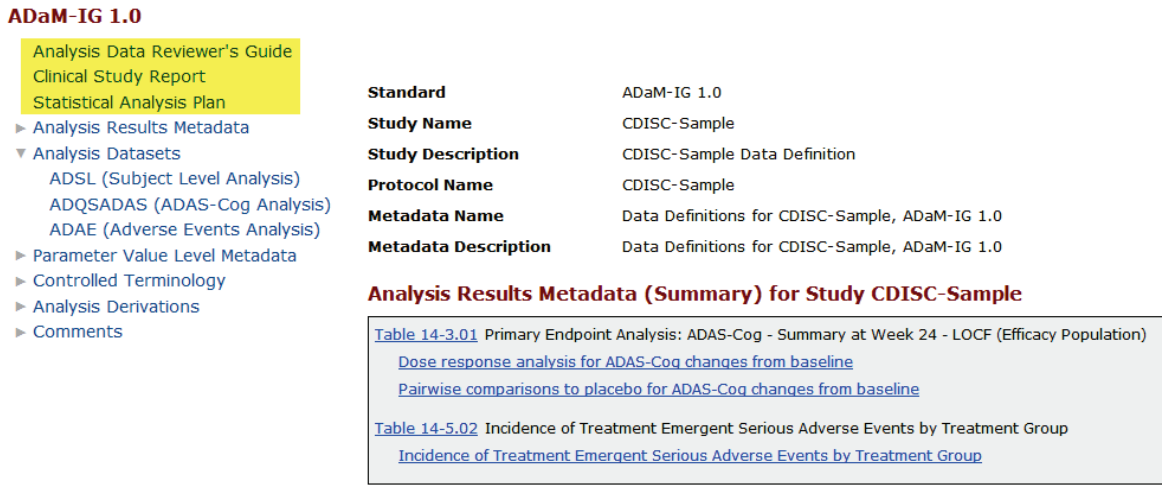


Figure 11 shows an example of the source_analysis data set. This data set contains metadata to support the Analysis Results Metadata extension to Define-XML version 2.0. The data set contains one record per analysis display per analysis result per analysis data set.

The CODE column in the source_analysisresults data set contains programming statements. Any whitespace (blanks) or carriage returns carriage returns ('0D'x) in in this column will be left unchanged by the SAS Clinical Standards Toolkit when creating the Define-XML file. An XSL stylesheet can then render this information in a formatted way when creating HTML. The SAS Clinical Standards Toolkit will convert "\n" to carriage returns ('0D'x) when creating the Define-XML file.

The stylesheet that comes with the Define-XML will display CODECONTEXT="`SAS version 9.2`" and CODE="`proc glm data = ADQSADAS;\n  where EFFFL='Y' and ANL01FL='Y' and AVISIT='Week 24' and PARAMCD="ACTOT";\n  class SITEGR1;\n  model CHG = TRTPN SITEGR1;\nrun;`" as follows:

Programming Statements	[SAS version 9.2]
	`proc glm data = ADQSADAS;`
	`  where EFFFL='Y' and ANL01FL='Y' and AVISIT='Week 24' and PARAMCD="ACTOT";`
	`  class SITEGR1;`
	`  model CHG = TRTPN SITEGR1;`
	`run;`

Figure 11 **Example of the source_analysisresults data set** (partial)

VIEWTABLE: Sampdata.Source_analysisresults (Source Analysis Results Metadata)

	DisplayIdentifier	DisplayName	DisplayDescription	ResultIdentifier	ParameterColumn	AnalysisReason	AnalysisPurpose	ResultDescription
1	RD.Table_14-3.01	Table 14-3.01	Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)	AR.Table_14-3.01.R.1	PARAMCD	SPECIFIED IN SAP	PRIMARY OUTCOME MEASURE	Dose response analysis for ADAS-Cog changes from baseline
2	RD.Table_14-3.01	Table 14-3.01	Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)	AR.Table_14-3.01.R.2	PARAMCD	SPECIFIED IN SAP	PRIMARY OUTCOME MEASURE	Pairwise comparisons to placebo for ADAS-Cog changes from baseline
3	RD.Table_14-5.02	Table 14-5.02	Incidence of Treatment Emergent Serious Adverse Events by Treatment Group	AR.Table_14-5.02.R.1		SPECIFIED IN SAP	PRIMARY OUTCOME MEASURE	Incidence of Treatment Emergent Serious Adverse Events by Treatment Group
4	RD.Table_14-5.02	Table 14-5.02	Incidence of Treatment Emergent Serious Adverse Events by Treatment Group	AR.Table_14-5.02.R.1		SPECIFIED IN SAP	PRIMARY OUTCOME MEASURE	Incidence of Treatment Emergent Serious Adverse Events by Treatment Group

VIEWTABLE: Sampdata.Source_analysisresults (Source Analysis Results Metadata)

	TableJoinComment	ResultDocumentation	CodeContext	Code	Table	AnalysisVariables	WhereClause
1		Linear model analysis of CHG for dose response; using randomized dose (0 for placebo; 54 for low dose; 81 for high dose) and site group in model. Used PROC GLM in SAS to produce p-value (from Type III SS for treatment dose). See page 4 (section 10.1.1) in the Statistical Analysis Plan.	SAS version 9.2	proc glm data = ADQSADAS;\n where EFFFL='Y' and ANL01FL='Y' and AVISIT='Week 24' and PARAMCD="ACTOT";\n class SITEGR1;\n model CHG = TRTPN SITEGR1;\nrun;	ADQSADAS	CHG	(PARAMCD EQ "ACTOT") AND (AVISIT EQ "Week 24") AND (EFFFL EQ "Y") AND (ANL01FL EQ "Y")
2		ANCOVA analysis of CHG performed to provide pairwise comparisons among treatment groups and adjusted means; using randomized treatment as class variable and site group as class variable in model and the baseline value as a covariate. See page 4 (section 10.1.1) in the Statistical Analysis Plan.	SAS version 9.2	proc glm data = ADQSADAS;\n where EFFFL='Y' and ANL01FL='Y' and AVISIT='Week 24' and PARAMCD="ACTOT";\n class TRTPN SITEGR1;\n model CHG = TRTPN SITEGR1 BASE;\n means TRTPN;\n lsmeans TRTPN / OM STDERR PDIFF CL;\nrun;	ADQSADAS	CHG	(PARAMCD EQ "ACTOT") AND (AVISIT EQ "Week 24") AND (EFFFL EQ "Y") AND (ANL01FL EQ "Y")
3	Get denominators for percentages from ADSL and counts and numerators from ADAE. Join ADAE with ADSL based on the unique subject identifier (USUBJID) keeping only records in ADAE for the numerator.	Unique count of subjects that experienced an Adverse Event by Preferred Term, System Organ Class, and Treatment Group and percentages based on the number of subjects in the safety population within each treatment group. The total number of times an event occurred was recorded by Preferred Term, System Organ Class, and Treatment Group. Fisher's exact test was used for treatment comparison of event rates. See page 5 (section 11.2) in the Statistical Analysis Plan.	SAS version 9.2		ADAE	AEBODSYS AEDECOD	(TRTEMFL EQ "Y") AND (AESER EQ "Y")
4	Get denominators for percentages from ADSL and counts and numerators from ADAE. Join ADAE with ADSL based on the unique subject identifier (USUBJID) keeping only records in ADAE for the numerator.	Unique count of subjects that experienced an Adverse Event by Preferred Term, System Organ Class, and Treatment Group and percentages based on the number of subjects in the safety population within each treatment group. The total number of times an event occurred was recorded by Preferred Term, System Organ Class, and Treatment Group. Fisher's exact test was used for treatment comparison of event rates. See page 5 (section 11.2) in the Statistical Analysis Plan.	SAS version 9.2		ADSL		SAFFL EQ "Y"

For analysis results based on a Basic Data Structure (BDS) the PARAMETERCOLUMN contains the name of the variable that contains the short name of the analysis parameter (PARAM). It is expected that the PARAM variable is also part of the WHERECLAUSE that specifies the selection criteria for the analysis subset. The WHERECLAUSE contains a complete list of the variables and their values used to identify the records selected for the analysis.

Appendix 2 contains the structure of the syntax of the WHERECLAUSE column as implemented by the SAS Clinical Standards Toolkit in a railroad diagram.

The following code can be used to create a Define-XML 2.0 file with the source metadata data sets as input:

```
%let _cstStandard=CDISC-DEFINE-XML;
%let _cstStandardVersion=2.0.0;

%cst_setStandardProperties(_cstStandard=CST-FRAMEWORK,_cstSubType=initialize);

%let studyRootPath=C:/cstSampleLibrary/cdisc-definexml-2.0.0-1.7;
%let studyOutputPath=C:/cstSampleLibrary/cdisc-definexml-2.0.0-1.7;
%let workPath=%sysfunc(pathname(work));


*******************************************************************************************;
* One strategy to defining the required library and file metadata for a CST process     *;
*  is to optionally build SASReferences in the WORK library.  An example of how to do   *;
*  this follows.                                                                        *;
*                                                                                       *;
* The call to cstutil_processsetup below tells CST how SASReferences will be provided   *;
*  and referenced.  If SASReferences is built in work, the call to cstutil processsetup *;
*  may, assuming all defaults, be as simple as:                                         *;
*        %cstutil_processsetup()                                                        *;
*******************************************************************************************;
%let _cstSetupSrc=SASREFERENCES;

%cst_createdsfromtemplate(
  _cstStandard=CST-FRAMEWORK, _cstType=control,
  _cstSubType=reference, _cstOutputDS=work.sasreferences
);

proc sql;
  insert into work.SASReferences
  values ("CST-FRAMEWORK"     "1.2"                     "messages"
          ""           "messages" "libref"  "input"
          "dataset"  "N"  "" ""  1  ""  "")
  values ("& cstStandard"     "& cstStandardVersion"    "messages"
          ""           "defmsg"   "libref"  "input"
          "dataset"  "N"  "" ""  2  ""  "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "autocall"
          ""           "auto1"    "fileref" "input"
          "folder"   "N"  "" ""  1  ""  "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "properties"
          "initialize" "inprop"   "fileref" "input"
          "file"     "N"  "" ""  1  ""  "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "results"
          "results"    "results"  "libref"  "output"
          "dataset"  "Y"  "" "&studyOutputPath/results"  .
          "definexml results adam.sas7bdat" "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "sourcemetadata"
          ""           "sampdata" "libref"  "input"
          "dataset"  "N"  "" "&studyRootPath/sascstdemodata/cdisc-adam-2.1/metadata" .
          "source_study.sas7bdat" "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "sourcedata"
          ""           "srcdata"  "libref"  "input"
          "folder"   "N"  "" "%sysfunc(pathname(work))"  . ""  "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "externalxml"
          "xml"        "extxml"   "fileref" "output"
          "file"     "Y"  "" "&studyOutputPath/sourcexml"  . "define-adam-2.1.xml" "")
  values ("&_cstStandard"     "&_cstStandardVersion"    "report"
          "outputfile" "html"     "fileref" "output"
          "file"     "Y"  ""  "&studyOutputPath/sourcexml" . " define-adam-2.1.html" "")
  values ("& cstStandard"     "& cstStandardVersion"    "referencexml"
          "stylesheet" "xslt01"   "fileref" "output"
          "file"     "Y"  ""   ""                          . "define2-0-0.xsl" "")
  ;
quit;

****************************************************;
* Process SASReferences file.                      *;
****************************************************;
%cstutil_processsetup();

****************************************************;
* Run the standard-specific Define-XML macros.     *;
****************************************************;
```

```
****************************************************;
* Read the source metadata                         *;
****************************************************;
%define_sourcetodefine(
  _cstOutLib=srcdata,
   cstSourceStudy=sampdata.source_study,
   cstSourceTables=sampdata.source_tables,
  _cstSourceColumns=sampdata.source_columns,
  _cstSourceCodeLists=sampdata.source_codelists,
   cstSourceDocuments=sampdata.source_documents,
  _cstSourceValues=sampdata.source_values,
  _cstSourceAnalysisResults=sampdata.source_analysisresults
  );

****************************************************;
* Create the Define-XML file                       *;
****************************************************;
%define_write(_cstCreateDisplayStyleSheet=1);

****************************************************;
* Validate the Define-XML file                     *;
****************************************************;
%cstutilxmlvalidate();

*****************************************************************************************;
* Create HTML rendition for browsers that do not allow local rendition of XSLT stylesheet *;
*****************************************************************************************;
proc xsl
  in=extxml xsl=xslt out=html;
run;
```

After running this program a results data set will be created which will show if there are any issues (results.definexml_results_adam, see Appendix 3). After the Define-XML file has been created, it can be reviewed by opening it in a browser. The browser will use the referenced stylesheet to render the XML file as HTML in the browser (Figure 12).

Figure 12 The Define-XML file displayed in a browser

ADaM-IG 1.0

- Analysis Data Reviewer's Guide
- Clinical Study Report
- Statistical Analysis Plan
- Analysis Results Metadata
- Analysis Datasets
- Parameter Value Level Metadata
- Controlled Terminology
- Analysis Derivations
- Comments

Date of Define-XML document generation: 2017-03-24T10:20:03-04:00
Define-XML version: 2.0.0
Stylesheet version: 2016-07-08

Standard	ADaM-IG 1.0
Study Name	CDISC-Sample
Study Description	CDISC-Sample Data Definition
Protocol Name	CDISC-Sample
Metadata Name	Data Definitions for CDISC-Sample, ADaM-IG 1.0
Metadata Description	Data Definitions for CDISC-Sample, ADaM-IG 1.0

Analysis Results Metadata - Summary

Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)
- Dose response analysis for ADAS-Cog changes from baseline
- Pairwise comparisons to placebo for ADAS-Cog changes from baseline

Table 14-5.02 Incidence of Treatment Emergent Serious Adverse Events by Treatment Group
- Incidence of Treatment Emergent Serious Adverse Events by Treatment Group

Analysis Results Metadata - Detail

Table 14-3.01

Display	**Table 14-3.01 [page=2] Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)**
Analysis Result	**Dose response analysis for ADAS-Cog changes from baseline**
Analysis Parameter(s)	PARAMCD = "ACTOT" (Adas-Cog(11) Subscore)
Analysis Variable(s)	CHG (Change from Baseline)
Analysis Reason	SPECIFIED IN SAP
Analysis Purpose	PRIMARY OUTCOME MEASURE
Data References (incl. Selection Criteria)	ADQSADAS [PARAMCD = "ACTOT" and AVISIT = "Week 24" and EFFFL = "Y" and ANL01FL = "Y"]
Documentation	Linear model analysis of CHG for dose response; using randomized dose (0 for placebo; 54 for low dose; 81 for high dose) and site group in model. Used PROC GLM in SAS to produce p-value (from Type III SS for treatment dose). See page 4 (section 10.1.1) in the Statistical Analysis Plan. Statistical Analysis Plan [page=4]

Programming Statements	[SAS version 9.2]
	`proc glm data = ADQSADAS;`
	`  where EFFFL='Y' and ANL01FL='Y' and AVISIT='Week 24' and PARAMCD="ACTOT";`
	`  class SITEGR1;`
	`  model CHG = TRTPN SITEGR1;`
	`run;`

CONCLUSION

SAS Clinical Standards Toolkit can create a complete Define-XML 2.0 file including Analysis Results Metadata in a convenient way without needing any XML knowledge based on 7 source metadata datasets.

REFERENCES

1. Lex Jansen (2012). Using the SAS® Clinical Standards Toolkit for define.xml creation. Proceedings of the Pharmaceutical Industry SAS® Users Group (PharmaSUG 2012, San Francisco, CA) (http://www.lexjansen.com/pharmasug/2012/HW/PharmaSUG-2012-HW02-SAS.pdf)

2. Lex Jansen (2013). Define-XML v2 – What's New. Proceedings of the 9th Pharmaceutical Users Software Exchange (PhUSE 2013, Brussels, Belgium) (http://www.lexjansen.com/phuse/2013/cd/CD05.pdf)

3. CDISC Define-XML Specification, Version 2.0, March 5, 2013 (http://www.cdisc.org/define-xml)

4. Case Report Tabulation Data Definition Specification (define.xml), Version 1.0, February 9, 2005 (http://www.cdisc.org/define-xml)

5. FDA Study Data Standards Catalog (Version 4.5; Effective 2016-08-31). (http://www.fda.gov/ForIndustry/DataStandards/StudyDataStandards/default.htm)

6. U.S. Department of Health and Human Services Food and Drug Administration Center for Drug Evaluation and Research (CDER), Center for Biologics Evaluation and Research (CBER). Study Data Technical Conformance Guide, Version 3.3, March 2017 (http://www.fda.gov/ForIndustry/DataStandards/StudyDataStandards/default.htm)

7. Department Of Health And Human Services Food and Drug Administration [Docket No. FDA–2014–N–1840] Electronic Study Data Submission; Data Standards; Support End Date for Case Report Tabulation Data Definition Specification Version 1.0, March 17, 2016 (http://www.gpo.gov/fdsys/pkg/FR-2016-03-17/pdf/2016-05958.pdf)

8. Benjamin P. Vali, Steve Wilson, Weiya Zhang (Office of Biostatistics (OB) FDA/CDER), and Susan Kenny (FDA/CDER Consultant) (2-15).ADaM Datasets for Review: An FDA Update. Pharmaceutical Users Software Exchange One Day Event (PhUSE SDE, April 30, 2015). (http://www.phusewiki.org/docs/2015_Chicago_SDE/FDA%20Update%20PhUSE_Chicago_30Apr2015_Print.pdf)

9. Pharmaceuticals and Medical Devices Agency, Technical Conformance Guide on Electronic Study Data Submissions, April 27, 2015 (Provisional Translation (as of July 2015)). (http://www.pmda.go.jp/english/review-services/reviews/advanced-efforts/0002.html)

10. Frank Dilorio and Jeffrey Abolifia (2016). Results-Level Metadata: What, How, and Why. Proceedings of the Pharmaceutical Industry SAS® Users Group (PharmaSUG 2016, Denver, CO, USA) (http://www.lexjansen.com/pharmasug/2016/DS/PharmaSUG-2016-DS03.pdf)

11. CDISC Analysis Data Model (ADaM), Version 2.1, December 17, 2009 (http://www.cdisc.org/adam)

12. SDTM/ADaM Pilot Projects (2007, 2009, 2013). Accessed on April 9, 2016. (http://www.cdisc.org/sdtmadam-pilot-project)

13. CDISC Analysis Results Metadata Specification Version 1.0 for Define-XML Version 2, January 27, 2015 (http://www.cdisc.org/adam)

14. CDISC Operational Data Model (ODM), Version 1.3.2, March 2013 (http://www.cdisc.org/odm)

15. Lex Jansen (2014). Creating Define-XML Version 2 with the SAS® Clinical Standards Toolkit 1.6. Proceedings of the Pharmaceutical Industry SAS® Users Group (PharmaSUG 2014, San Diego, CA, USA) (http://www.lexjansen.com/pharmasug/2014/DS/PharmaSUG-2014-DS23-SAS.pdf)

16. SAS Institute Inc. 2016. SAS® Clinical Standards Toolkit 1.7.1. Cary, NC: SAS Institute Inc. (http://support.sas.com/rnd/base/cdisc/cst/index.html)

17. SAS Technical Support Usage Note 60134: SAS® Clinical Standards Toolkit 1.7.1 - MethodDef element has some content that is not reflected in the source_columns and source_values metadata. (http://support.sas.com/kb/60/134.html)

18. Analysis Data Reviewer's Guide (ADRG), work in progress, as of April 23, 2014 (http://www.phusewiki.org/wiki/index.php?title=Analysis_Data_Reviewer%27s_Guide)

19. Terminology Resources: NCI Enterprise Vocabulary Services (EVS), Dictionaries, and Partnerships (http://www.cancer.gov/cancertopics/cancerlibrary/terminologyresources/cdisc)

CONTACT INFORMATION

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

Lex Jansen

SAS Institute Inc.
Email: lex.jansen@sas.com

APPENDIX 1: SOURCE METADATA TABLE STRUCTURES

source_study

#	Column Name	Column Type	Column Length	Column Label
1	sasref	char	8	SASreferences sourcedata libref
2	studyname	char	128	Short external name for the study
3	studydescription	char	2000	Description of the study
4	protocolname	char	128	Sponsors internal name for the protocol
5	formalstandardname	char	2000	Formal Name of Standard
6	formalstandardversion	char	2000	Formal Version of Standard
7	studyversion	char	128	Unique study version identifier
8	standard	char	20	Name of Standard
9	standardversion	char	20	Version of Standard

source_tables

#	Column Name	Column Type	Column Length	Column Label
1	sasref	char	8	SASreferences sourcedata libref
2	table	char	32	Table Name
3	label	char	200	Table Label
4	order	num	8	Table order
5	domain	char	32	Domain
6	domaindescription	char	256	Domain description
7	class	char	40	Observation Class within Standard
8	xmlpath	char	200	(Relative) path to xpt file
9	xmltitle	char	200	Title for xpt file
10	structure	char	200	Table Structure
11	purpose	char	10	Purpose
12	keys	char	200	Table Keys
13	state	char	20	Data Set State (Final, Draft)
14	date	char	20	Release Date
15	comment	char	1000	Comment
16	studyversion	char	128	Unique study version identifier
17	standard	char	20	Name of Standard
18	standardversion	char	20	Version of Standard

source_columns[1]				
#	**Column Name**	**Column Type**	**Column Length**	**Column Label**
1	sasref	char	8	SASreferences sourcedata libref
2	table	char	32	Table Name
3	column	char	32	Column Name
4	label	char	200	Column Description
5	order	num	8	Column Order
6	type	char	1	Column Type
7	length	num	8	Column Length
8	displayformat	char	200	Display Format
9	significantdigits	num	8	Significant Digits
10	xmldatatype	char	18	XML Data Type
11	xmlcodelist	char	128	SAS Format/XML Codelist
12	core	char	10	Column Required or Optional
13	origin	char	40	Column Origin
14	origindescription	char	1000	Column Origin Description
15	role	char	200	Column Role
16	algorithm	char	1000	Computational Algorithm or Method
17	algorithmtype	char	11	Type of Algorithm
18	formalexpression	char	1000	Formal Expression for Algorithm
19	formalexpressioncontext	char	1000	Context to be used when evaluating the FormalExpression content
20	comment	char	1000	Comment
21	studyversion	char	128	Unique study version identifier
22	standard	char	20	Name of Standard
23	standardversion	char	20	Version of Standard

[1] The algorithmtype, formalexpression, and formalexpressioncontext variables were intially not part of the source_columns data set in SAS Clinical Standard Toolkit (1.7.1). SAS Technical Support Usage Note 60134 [17] shows how to add these variables to the source_columns data set.

source_codelists				
#	**Column Name**	**Column Type**	**Column Length**	**Column Label**
1	sasref	char	8	SASreferences sourcedata libref
2	codelist	char	128	Unique identifier for this CodeList
3	codelistname	char	128	CodeList Name
4	codelistdescription	char	2000	CodeList Description
5	codelistncicode	char	10	Codelist NCI Code
6	codelistdatatype	char	7	CodeList item value data type (integer | float | text | string)
7	sasformatname	char	32	SAS format name
8	codedvaluechar	char	512	Value of the codelist item (character)
9	codedvaluenum	num	8	Value of the codelist item (numeric)
10	decodetext	char	2000	Decode value of the codelist item
11	decodelanguage	char	17	Language
12	codedvaluencicode	char	10	Codelist Item NCI Code
13	rank	num	8	CodedValue order relative to other item values
14	ordernumber	num	8	Display order of the item within the CodeList.
15	extendedvalue	char	3	Coded value that has been used to extend external controlled terminology
16	dictionary	char	200	Name of the external codelist
17	version	char	200	Version designator of the external codelist
18	ref	char	512	Reference to a local instance of the dictionary
19	href	char	512	URL of an external instance of the dictionary
20	studyversion	char	128	Unique study version identifier
21	standard	char	20	Name of Standard
22	standardversion	char	20	Version of Standard

source_values[2]				
#	**Column Name**	**Column Type**	**Column Length**	**Column Label**
1	sasref	char	8	SASreferences sourcedata libref
2	table	char	32	Table Name
3	column	char	32	Column Name
4	whereclause	char	1000	Where Clause
5	whereclausecomment	char	1000	Where Clause comment
6	label	char	200	Column Description
7	order	num	8	Column Order
8	type	char	1	Column Type
9	length	num	8	Column Length
10	displayformat	char	200	Display Format
11	significantdigits	num	8	Significant Digits
12	xmldatatype	char	18	XML Data Type
13	xmlcodelist	char	128	SAS Format/XML Codelist
14	core	char	10	Column Required or Optional
15	origin	char	40	Column Origin
16	origindescription	char	1000	Column Origin Description
17	role	char	200	Column Role
18	algorithm	char	1000	Computational Algorithm or Method
19	algorithmtype	char	11	Type of Algorithm
20	formalexpression	char	1000	Formal Expression for Algorithm
21	formalexpressioncontext	char	1000	Context to be used when evaluating the FormalExpression content
22	comment	char	1000	Comment
23	studyversion	char	128	Unique study version identifier
24	standard	char	20	Name of Standard
25	standardversion	char	20	Version of Standard

[2] The algorithmtype, formalexpression, and formalexpressioncontext variables were intially not part of the source_values data set in SAS Clinical Standard Toolkit (1.7.1). SAS Technical Support Usage Note 60134 [17] shows how to add these variables to the source_values data set.

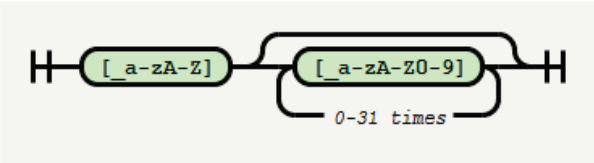
source_documents				
#	**Column Name**	**Column Type**	**Column Length**	**Column Label**
1	sasref	char	8	SASreferences sourcedata libref
2	doctype	char	10	Document Type
3	href	char	512	The pathname and filename of the target dataset relative to the define.xml
4	title	char	2000	Meaningful description, label, or location of the document leaf
5	pdfpagereftype	char	16	Type of Page Reference (PhysicalRef/NamedDestination)
6	pdfpagerefs	char	200	Page Reference
7	table	char	32	Table Name
8	column	char	32	Column Name
9	whereclause	char	1000	Where Clause
10	displayidentifier	char	128	Analysis Display Identifier
11	resultidentifier	char	128	Analysis Display Result Identifier
12	studyversion	char	128	Unique study version identifier
13	standard	char	20	Name of Standard
14	standardversion	char	20	Version of Standard

source_analysisresults				
#	**Column Name**	**Column Type**	**Column Length**	**Column Label**
1	sasref	char	8	SASreferences sourcedata libref
2	displayidentifier	char	128	Unique identifier for analysis display
3	displayname	char	2000	Title of display
4	displaydescription	char	2000	Description of display
5	resultidentifier	char	128	Specific analysis result within display
6	resultdescription	char	2000	Description of analysis result within display
7	parametercolumn	char	8	Name of the column that holds the parameter
8	analysisreason	char	2000	Reason for analysis
9	analysispurpose	char	2000	Purpose of analysis
10	tablejoincomment	char	2000	Comment describing how to join tables
11	resultdocumentation	char	2000	Documentation of analysis result within display
12	codecontext	char	128	Name and version of computer language
13	code	char	2000	Programming statements
14	table	char	32	Table Name
15	analysisvariables	char	1024	Analysis Variable List
16	whereclause	char	1000	Where Clause
17	studyversion	char	128	Unique Study Version Identifier
18	standard	char	20	Name of Standard
19	standardversion	char	20	Version of Standard

APPENDIX 2: RAILROAD DIAGRAM DEFINING THE WHERECLAUSE SYNTAX

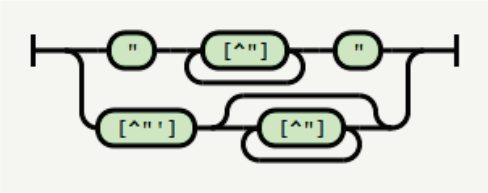
DOMAIN VARIABLE

A letter or underscore followed by 0 to 31 letters, digits or underscores
Notice that CDISC currently has stricter rules, since it only allows 8 characters.

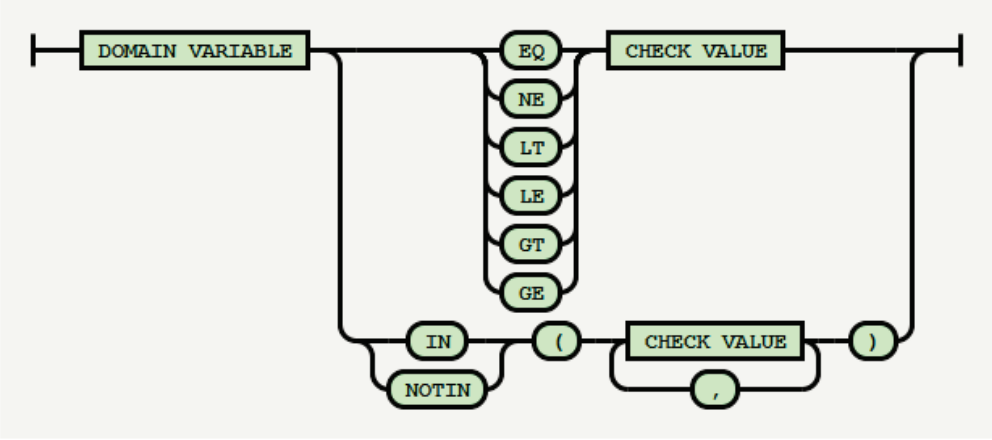


CHECK VALUE

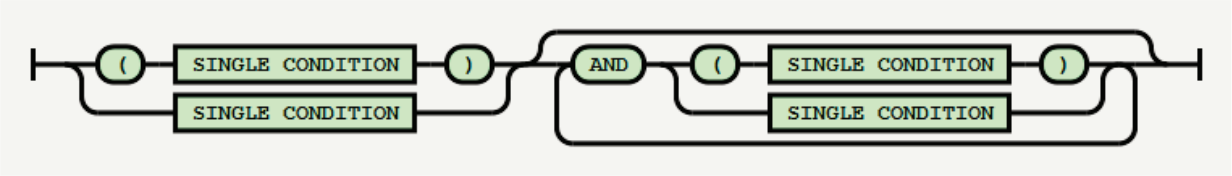
A non-empty string without a double quote enclosed in double quotes, or
a character that is not a single or double quote, followed by a string without a double quote.
The latter includes the representation of numbers.



SINGLE CONDITION



WHERECLAUSE



This diagram was created with the Railroad Diagram Generator, a small JS+SVG library for drawing railroad syntax diagrams by Tab Atkins Jr., which can be found online at https://github.com/tabatkins/railroad-diagrams.

APPENDIX 3: RESULTS DATA SET

	resultid	srcdata	message	resultseverity
1	CST0108	CST_SETPROPERTIES	The properties were processed from the PATH c:\cstGlobalLibrary/standards/cst-framework-1.7/programs/initialize.properties	Info
2	CST0200	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to c:\cstGlobalLibrary/standards/cst-framework-1.7/templates to perform the template lookup	Info
3	CST0102	CST_CREATEDSFROMTEMPLATE	work.sasreferences (SAS File and Library References) was created as requested	Info
4	CST0200	CSTUTIL_PROCESSSETUP	Process setup is using this SASReferences: C:\Users\frjans\AppData\Local\Temp\SAS Temporary Files_TD1928_L77359_/sasreferences	Info
5	CST0200	CST_INSERTSTANDARDSASREFS	SASReferences data set was successfully validated	Info
6	CST0200	CSTUTIL_ALLOCATESASREFERENCES	SASReferences data set was successfully validated	Info
7	CST0108	CST_SETPROPERTIES	The properties were processed from the PATH c:\cstGlobalLibrary/standards/cdisc-definexml-2.0.0-1.7/programs/initialize.properties	Info
8	CST0200	DEFINE_SOURCETODEFINE	PROCESS STANDARD: CDISC-DEFINE-XML	Info
9	CST0200	DEFINE_SOURCETODEFINE	PROCESS STANDARDVERSION: 2.0.0	Info
10	CST0200	DEFINE_SOURCETODEFINE	PROCESS DRIVER: create_definexml_from_source_adam.sas	Info
11	CST0200	DEFINE_SOURCETODEFINE	PROCESS DATE: 2016-02-24T12:57:43	Info
12	CST0200	DEFINE_SOURCETODEFINE	PROCESS TYPE: SOURCE TO DEFINE-XML	Info
13	CST0200	DEFINE_SOURCETODEFINE	PROCESS SASREFERENCES: C:\Users\frjans\AppData\Local\Temp\SAS Temporary Files_TD1928_L77359_/sasreferences	Info
14	CST0200	DEFINE_SOURCETODEFINE	PROCESS STUDYROOTPATH: c:\cstSampleLibrary/cdisc-definexml-2.0.0-1.7/sascstdemodata	Info
15	CST0200	DEFINE_SOURCETODEFINE	PROCESS STUDYOUTPUTPATH: c:\cstSampleLibrary/cdisc-definexml-2.0.0-1.7	Info
16	CST0200	DEFINE_SOURCETODEFINE	PROCESS GLOBALLIBRARY: c:\cstGlobalLibrary	Info
17	CST0200	DEFINE_SOURCETODEFINE	PROCESS CSTVERSION: 1.7	Info
18	CST0122	CST_CREATETABLESFORDATASTANDARD	The tables were created for CDISC-DEFINE-XML 2.0.0 in library srcdata	Info
19	CST0102	DEFINE_SOURCESTUDY	srcdata.DefineDocument was created as requested (1 record)	Info
20	CST0102	DEFINE_SOURCESTUDY	srcdata.Study was created as requested (1 record)	Info
21	CST0102	DEFINE_SOURCESTUDY	srcdata.MetaDataVersion was created as requested (1 record)	Info
22	CST0102	DEFINE_SOURCETABLES	srcdata.ItemGroupDefs was created as requested (3 records)	Info
23	CST0102	DEFINE_SOURCETABLES	srcdata.CommentDefs was created as requested (3 records)	Info
24	CST0102	DEFINE_SOURCETABLES	srcdata.TranslatedText was created as requested (3 records)	Info
25	CST0102	DEFINE_SOURCETABLES	srcdata.TranslatedText was created as requested (6 records)	Info
26	CST0102	DEFINE_SOURCETABLES	srcdata.Aliases was created as requested (0 records)	Info
27	CST0102	DEFINE_SOURCETABLES	srcdata.ItemGroupLeaf was created as requested (3 records)	Info
28	CST0102	DEFINE_SOURCETABLES	srcdata.ItemGroupLeafTitles was created as requested (3 records)	Info
29	CST0102	DEFINE_SOURCECOLUMNS	srcdata.ItemDefs was created as requested (143 records)	Info
30	CST0102	DEFINE_SOURCECOLUMNS	srcdata.MethodDefs was created as requested (54 records)	Info
31	CST0102	DEFINE_SOURCECOLUMNS	srcdata.CommentDefs was created as requested (15 records)	Info
32	CST0102	DEFINE_SOURCECOLUMNS	srcdata.ItemGroupItemRefs was created as requested (143 records)	Info
33	CST0102	DEFINE_SOURCECOLUMNS	srcdata.TranslatedText was created as requested (149 records)	Info
34	CST0102	DEFINE_SOURCECOLUMNS	srcdata.TranslatedText was created as requested (161 records)	Info
35	CST0102	DEFINE_SOURCECOLUMNS	srcdata.TranslatedText was created as requested (215 records)	Info
36	CST0102	DEFINE_SOURCECOLUMNS	srcdata.TranslatedText was created as requested (288 records)	Info
37	CST0102	DEFINE_SOURCECOLUMNS	srcdata.ItemOrigin was created as requested (140 records)	Info
38	CST0102	DEFINE_SOURCECODELISTS	srcdata.CodeLists was created as requested (29 records)	Info
39	CST0102	DEFINE_SOURCECODELISTS	srcdata.TranslatedText was created as requested (317 records)	Info
40	CST0102	DEFINE_SOURCECODELISTS	srcdata.EnumeratedItems was created as requested (98 records)	Info
41	CST0102	DEFINE_SOURCECODELISTS	srcdata.CodeListItems was created as requested (97 records)	Info
42	CST0102	DEFINE_SOURCECODELISTS	srcdata.TranslatedText was created as requested (414 records)	Info
43	CST0102	DEFINE_SOURCECODELISTS	srcdata.ExternalCodeLists was created as requested (1 record)	Info
44	CST0102	DEFINE_SOURCECODELISTS	srcdata.Aliases was created as requested (12 records)	Info
45	CST0102	DEFINE_SOURCECODELISTS	srcdata.Aliases was created as requested (32 records)	Info
46	CST0102	DEFINE_SOURCECODELISTS	srcdata.Aliases was created as requested (45 records)	Info
47	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	Validating WhereClause in sampdata.source_values (creating _cstSourceValues1_1376)	Info
48	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	There were no issues with WhereClause in sampdata.source_values	Info
49	CST0102	DEFINE_SOURCEVALUES	srcdata.ItemDefs was created as requested (149 records)	Info
50	CST0102	DEFINE_SOURCEVALUES	srcdata.MethodDefs was created as requested (56 records)	Info
51	CST0102	DEFINE_SOURCEVALUES	srcdata.CommentDefs was created as requested (18 records)	Info
52	CST0102	DEFINE_SOURCEVALUES	srcdata.ValueLists was created as requested (3 records)	Info
53	CST0102	DEFINE_SOURCEVALUES	srcdata.ValueListItemRefs was created as requested (6 records)	Info
54	CST0102	DEFINE_SOURCEVALUES	srcdata.ItemValueListRefs was created as requested (3 records)	Info
55	CST0102	DEFINE_SOURCEVALUES	srcdata.ItemRefWhereClauseRefs was created as requested (6 records)	Info
56	CST0102	DEFINE_SOURCEVALUES	srcdata.WhereClauseDefs was created as requested (6 records)	Info
57	CST0102	DEFINE_SOURCEVALUES	srcdata.WhereClauseRangeChecks was created as requested (6 records)	Info
58	CST0102	DEFINE_SOURCEVALUES	srcdata.WhereClauseRangeCheckValues was created as requested (19 records)	Info
59	CST0102	DEFINE_SOURCEVALUES	srcdata.TranslatedText was created as requested (420 records)	Info
60	CST0102	DEFINE_SOURCEVALUES	srcdata.TranslatedText was created as requested (423 records)	Info
61	CST0102	DEFINE_SOURCEVALUES	srcdata.TranslatedText was created as requested (425 records)	Info
62	CST0102	DEFINE_SOURCEVALUES	srcdata.TranslatedText was created as requested (426 records)	Info
63	CST0102	DEFINE_SOURCEVALUES	srcdata.ItemOrigin was created as requested (146 records)	Info
64	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	Validating WhereClause in _cstSourceAnalysisResults_3900 (creating _cstSourceAnalysisResults_3900)	Info
65	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	There were no issues with WhereClause in _cstSourceAnalysisResults_3900	Info
66	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisresultdisplays was created as requested (2 records)	Info
67	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisresults was created as requested (3 records)	Info
68	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisdatasets was created as requested (3 records)	Info
69	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisdataset was created as requested (4 records)	Info
70	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysiswhereclauserefs was created as requested (4 records)	Info
71	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisvariables was created as requested (4 records)	Info
72	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisdocumentation was created as requested (3 records)	Info
73	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.analysisprogrammingcode was created as requested (3 records)	Info
74	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.WhereClauseDefs was created as requested (10 records)	Info
75	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.WhereClauseRangeChecks was created as requested (17 records)	Info
76	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.WhereClauseRangeCheckValues was created as requested (30 records)	Info
77	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.TranslatedText was created as requested (428 records)	Info
78	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.TranslatedText was created as requested (431 records)	Info
79	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.TranslatedText was created as requested (434 records)	Info
80	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.TranslatedText was created as requested (437 records)	Info

81	CST0102	DEFINE_SOURCEANALYSISRESULTS	srcdata.CommentDefs was created as requested (19 records)	Info
82	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	Validating WhereClause in sampdata.source_documents (creating _cstSourceDocuments_2569)	Info
83	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	There were no issues with WhereClause in sampdata.source_documents	Info
84	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	Validating WhereClause in sampdata.source_values (creating _cstSourceValues1_2569)	Info
85	DEF0097	DEFINEUTIL_VALIDATEWHERECLAUSE	There were no issues with WhereClause in sampdata.source_values	Info
86	CST0102	DEFINE_SOURCEDOCUMENTS	srcdata.AnnotatedCRFs was created as requested (0 records)	Info
87	CST0102	DEFINE_SOURCEDOCUMENTS	srcdata.SupplementalDocs was created as requested (3 records)	Info
88	CST0102	DEFINE_SOURCEDOCUMENTS	srcdata.MDVLeaf was created as requested (8 records)	Info
89	CST0102	DEFINE_SOURCEDOCUMENTS	srcdata.MDVLeafTitles was created as requested (8 records)	Info
90	CST0102	DEFINE_SOURCEDOCUMENTS	srcdata.DocumentRefs was created as requested (14 records)	Info
91	CST0102	DEFINE_SOURCEDOCUMENTS	srcdata.PDFPageRefs was created as requested (8 records)	Info
92	CST0102	CSTUTIL_SAVERESULTS	results.sourcetodefinexml_adam_results was created as requested	Info
93	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS STANDARD:	Info
94	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS STANDARDVERSION:	Info
95	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS DRIVER: create_definexml_from_source_adam.sas	Info
96	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS DATE: 2016-02-24T12:57:49	Info
97	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS TYPE: FILEIO	Info
98	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS SASREFERENCES: C:\Users\frjans\AppData\Local\Temp\SAS Temporary Files_TD1928_L77359_/_cstsasrefs.sas7bdat	Info
99	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS STUDYROOTPATH: c:\cstSampleLibrary/cdisc-definexml-2.0.0-1.7/sascstdemodata	Info
100	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS STUDYOUTPUTPATH: c:\cstSampleLibrary/cdisc-definexml-2.0.0-1.7	Info
101	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS GLOBALLIBRARY: c:\cstGlobalLibrary	Info
102	CST0200	CSTUTIL_ALLOCATESASREFERENCES	PROCESS CSTVERSION: 1.7	Info
103	CST0122	CST_CREATETABLESFORDATASTANDARD	The tables were created for CDISC-DEFINE-XML 2.0.0 in library _cst7298	Info
104	CST0200	JAVA CHECK	No Java issues	Info
105	CST0200	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to c:\cstGlobalLibrary/standards/cst-framework-1.7/templates to perform the template lookup	Info
106	CST0102	CST_CREATEDSFROMTEMPLATE	work._cstxmllog (Results) was created as requested	Info
107	CST0191	XML TRANSFORMER	Transform starting.	Info
108	CST0191	XML TRANSFORMER	Using JRE: C:\PROGRA~1\SASHome\SASPRI~1\9.4\jre	Info
109	CST0191	XML TRANSFORMER PARAMETER	Import Or Export: EXPORT	Info
110	CST0191	XML TRANSFORMER PARAMETER	Standards XML Path: c:/cstSampleLibrary/cdisc-definexml-2.0.0-1.7/sourcexml/define-adam-2.1.xml	Info
111	CST0191	XML TRANSFORMER PARAMETER	Fail on Validation Error: false	Info
112	CST0191	XML TRANSFORMER PARAMETER	Standard Name: CDISC-DEFINE-XML	Info
113	CST0191	XML TRANSFORMER PARAMETER	Standard Version: 2.0.0	Info
114	CST0191	XML TRANSFORMER PARAMETER	Schema Repository Location: c:/cstGlobalLibrary/schema-repository	Info
115	CST0191	XML TRANSFORMER PARAMETER	XSL Repository Location: c:/cstGlobalLibrary/xsl-repository	Info
116	CST0191	XML TRANSFORMER PARAMETER	Output Encoding: UTF-8	Info
117	CST0191	XML TRANSFORMER PARAMETER	Log File Location: C:/Users/frjans/AppData/Local/Temp/SAS Temporary Files/_TD1928_L77359_/_log4919_write.xml	Info
118	CST0191	XML TRANSFORMER PARAMETER	Header Comment Text: Produced from SAS data using the SAS Clinical Standards Toolkit 1.7	Info
119	CST0191	XML TRANSFORMER PARAMETER	Is Validating XML: true	Info
120	CST0191	XML TRANSFORMER PARAMETER	Creating Display Stylesheet: true	Info
121	CST0191	XML TRANSFORMER PARAMETER	Custom Stylesheet: c:/cstGlobalLibrary/standards/cdisc-definexml-2.0.0-1.7/stylesheet/define2-0-0.xsl	Info
122	CST0191	XML TRANSFORMER PARAMETER	Custom Stylesheet Output Shortname: define2-0-0.xsl	Info
123	CST0191	XML TRANSFORMER PARAMETER	Creating Output Folders: true	Info
124	CST0191	XML TRANSFORMER	Transform complete.	Info
125	CST0191	XML TRANSFORMER	Transform time: 1125 ms.	Info
126	CST0191	XML TRANSFORMER PARAMETER	XML File to Validate: c:/cstSampleLibrary/cdisc-definexml-2.0.0-1.7/sourcexml/define-adam-2.1.xml	Info
127	CST0191	XML TRANSFORMER PARAMETER	Schema being validated against: c:/cstGlobalLibrary/schema-repository/cdisc-arm-1.0/arm1-0-0.xsd	Info
128	CST0191	XML TRANSFORMER	The document validated successfully	Info
129	DEF0010	DEFINE_WRITE	The DEFINE-XML file was created at c:\cstSampleLibrary\cdisc-definexml-2.0.0-1.7\sourcexml\define-adam-2.1.xml	Info
130	CST0102	CSTUTIL_SAVERESULTS	results.sourcetodefinexml_adam_results was created as requested	Info
131	CST0200	CSTUTILXMLVALIDATE	Starting XML Validation	Info
132	CST0200	JAVA CHECK	No Java issues	Info
133	CST0200	CST_CREATEDSFROMTEMPLATE	The SAS libref csttmplt was allocated to c:\cstGlobalLibrary/standards/cst-framework-1.7/templates to perform the template lookup	Info
134	CST0102	CST_CREATEDSFROMTEMPLATE	work._cstxmllog (Results) was created as requested	Info
135	CST0191	XML TRANSFORMER	Transform starting.	Info
136	CST0191	XML TRANSFORMER	Using JRE: C:\PROGRA~1\SASHome\SASPRI~1\9.4\jre	Info
137	CST0191	XML TRANSFORMER PARAMETER	Import Or Export: EXPORT	Info
138	CST0191	XML TRANSFORMER PARAMETER	Standards XML Path: c:/cstSampleLibrary/cdisc-definexml-2.0.0-1.7/sourcexml/define-adam-2.1.xml	Info
139	CST0191	XML TRANSFORMER PARAMETER	Fail on Validation Error: false	Info
140	CST0191	XML TRANSFORMER PARAMETER	Standard Name: CDISC-DEFINE-XML	Info
141	CST0191	XML TRANSFORMER PARAMETER	Standard Version: 2.0.0	Info
142	CST0191	XML TRANSFORMER PARAMETER	Schema Repository Location: c:\cstGlobalLibrary/schema-repository	Info
143	CST0191	XML TRANSFORMER PARAMETER	XSL Repository Location: null	Info
144	CST0191	XML TRANSFORMER PARAMETER	Output Encoding: UTF-8	Info
145	CST0191	XML TRANSFORMER PARAMETER	Log File Location: C:\Users\frjans\AppData\Local\Temp\SAS Temporary Files_TD1928_L77359_/_log9387_xmlvalidate.xml	Info
146	CST0191	XML TRANSFORMER PARAMETER	Header Comment Text: Produced from SAS data using the SAS Clinical Standards Toolkit	Info
147	CST0191	XML TRANSFORMER PARAMETER	Is Validating XML: true	Info
148	CST0191	XML TRANSFORMER PARAMETER	Creating Display Stylesheet: false	Info
149	CST0191	XML TRANSFORMER PARAMETER	Custom Stylesheet: null	Info
150	CST0191	XML TRANSFORMER PARAMETER	Custom Stylesheet Output Shortname: null	Info
151	CST0191	XML TRANSFORMER PARAMETER	Creating Output Folders: true	Info
152	CST0191	XML TRANSFORMER PARAMETER	XML File to Validate: c:/cstSampleLibrary/cdisc-definexml-2.0.0-1.7/sourcexml/define-adam-2.1.xml	Info
153	CST0191	XML TRANSFORMER PARAMETER	Schema being validated against: c:\cstGlobalLibrary/schema-repository/cdisc-arm-1.0/arm1-0-0.xsd	Info
154	CST0191	XML TRANSFORMER	The document validated successfully	Info
155	CST0100	CSTUTIL_APPENDRESULTDS	No errors detected in the XML file	Info
156	CST0102	CSTUTIL_SAVERESULTS	results.sourcetodefinexml_adam_results was created as requested	Info