CDISC Define-XML Specification

Version 2.1 (Final)

Prepared by the
CDISC Define-XML Team

[redacted]

Notes to Readers

- This is the specification for Version 2.1 of the CDISC Define-XML standard.
- This is an update of the CDISC Define-XML Version 2.0 Specification.

Revision History

Date	Version	Summary of Changes
2019-05-15	2.1	Production version of the Define-XML Version 2.1 Specification
2018-08-04	2.1 DRAFT 2	Updated review DRAFT of Version 2.1 of the Define-XML standard.
2016-11-04	2.1 DRAFT	Preliminary review DRAFT of Version 2.1 of the Define-XML standard
2013-03-05	2.0.0	Production version of the Define-XML Version 2.0 Specification
2012-08-31	2.0 DRAFT	Public review DRAFT of Version 2.0 of the Define-XML standard
2012-07-23	2.0.0.3	Update for SRC review and draft publication
2011-09-16	2.0.0.2	Updated Define-XML 2.0.0 specification in new format
2009-11-27	2.0.0.1	Draft specification for Define-XML 2.0.0
2005-02-09	1.0.0	Administrative update
2005-02-05	1.0.0	This is the official implementation version of the Case Report Tabulation Data Definition specification.

CONTENTS

1 Introduction

1.1 Purpose of This Document

This specification document describes Define-XML 2.1, a metadata standard used to describe any tabular dataset structure. The primary use case for Define-XML is to describe CDISC Study Data Tabulation Model (SDTM), Standard for Exchange of Nonclinical Data (SEND), and Analysis Data Model (ADaM) datasets for the purpose of submissions to regulatory authorities. However, Define-XML also serves as a metadata exchange mechanism for other parties seeking to exchange CDISC-modeled dataset structures as well as proprietary (non-CDISC) dataset structures. For dataset structures modeled following CDISC standards, refer to the SDTM, ADaM, and SEND Implementation Guides (IG); see Section 2.2, References.

The Define-XML model is implemented using extensions to the CDISC Operational Data Model (ODM) XML schema. The intent is to comply fully with all applicable regulations and guidance. This document includes substantially all of the material from the Define-XML v2.0 specification in addition to Define-XML v2.1 changes that include an updated approach to def:Origin, versioning of standards and controlled terminology, improved SENDIG support, v2.0 errata fixes, and additional updates (see Section 1.1.3, Relationship to Prior Define-XML Specifications).

Define-XML v.1.0 described the requirements for constructing Define-XML documents to replace the define.pdf documents recommended in the FDA’s 1999 *Regulatory Submissions in Electronic Format, General Considerations* guidance (see Section 2.2, References). Since that time, Define-XML documents have proven to be a useful mechanism for transmitting case report tabulation (CRT) metadata. One of the key benefits to FDA reviewers is that this format provides both a machine-readable format for use by the various FDA software applications and, through the provision of an XSL stylesheet, a browser-based report describing the contents of a study.

1.1.1 Organization of This Document

This document is organized into the following sections:

- Section 1, Introduction, includes the overall purpose of the document, and discusses how Define-XML fits within the CDISC standards framework.
- Section 2, Abbreviations and References
- Section 3, Conformity and General Issues
- Section 4, General Specifications for Define-XML, describes how to create a Define-XML document from a business perspective. Information in this section illustrates how Define-XML is used to represent certain types of metadata for the CDISC SDTM, SEND, and ADaM standards.
- Section 5, Specification, provides an element-by-element technical specification. The content of this section is normative.
- Section 6, Global Element Ordering, provides a diagram to illustrate the required structure of a Define-XML document.
- Appendices

1.1.2 How to Read This Specification

The Define-XML 2.1 specification is best read online so that the reader can benefit from the hyperlinks to both internal and external references.

The following guidelines may be helpful in reading this document:

1. First, read Section 4, General Specifications for Define-XML, to gain a general understanding of how Define-XML represents metadata for SDTM, SEND, and ADaM data. This section includes many examples of the XML fragments used to represent metadata for identifying CDISC standards, datasets, variables, codelists, and value-level metadata.

2. Read Section 1.1.3, Relationship to Prior Define-XML Specifications, to find out about new features and changes in Version 2.1.
3. Browse through Section 5, Specification, for more detailed technical information. This section serves as a technical reference guide for Define-XML users who wish to implement the standard. It provides specifications for each of the XML elements and attributes that are used in Define-XML documents for this version of the standard and supplements the syntactic information provided by the Define-XML schema with business rules.
4. Following the links in Section 2.2, References, read the implementation guides for the versions of SDTM, SEND or ADaM datasets you wish to describe with your Define-XML document. Review the controlled terminology publications used by the SDTM, SEND, or ADaM datasets you will describe in your Define-XML document.

1.1.3 Relationship to Prior Define-XML Specifications

This document includes changes to the Define-XML Version 2.0 specification that was published in 2013. The major changes between versions 2.0 and 2.1 are:

- CDISC content standards used within a Define-XML document may be identified and referenced by dataset definitions.
- CDISC controlled terminology publications used within a Define-XML document may be identified and referenced by codelist definitions.
- Representation of SDTM Origin metadata has been enhanced to identify the source in addition to the origin details and to support multiple origins.
- The dataset Class attribute has been re-implemented as a child element with support for definition of Subclass.
- A def:Context attribute has been added to the ODM root element to indicate that Define-XML is for a regulatory submission.
- An optional def:DocumentRef/def:PDFPageRef/@Title attribute was added to allow a more specific and descriptive reference to a page link.
- Added support for providing SAS names longer than 8 characters to support legacy dataset submissions using the Alias element.
- New examples have been provided for both SDTM and ADaM Define-XML.
- Enumerations have been added to the XML Schema to control values for the elements and attributes listed below.

Define-XML Entities with Schema Enumerations
def:Standard@Name
def:Standard@Type
def:Standard@Status
def:Standard/@PublishingSet
ItemGroupDef/@def:Class
def:Origin/@Type
def:Origin/@Source
def:PDFPageRef/@Type

1.2 Electronic Submission Background

The Define-XML standard supports a wide range of uses, including data submissions to regulatory authorities. In countries such as the United States and Japan, where the approval process for regulated human and animal health products may require the submission of data from clinical trials or other studies, Define-XML is among the standards that may be required as part of a complete data submission. In such cases, a technical conformance guide published by the relevant authority will typically list these standards and reference a data standards catalog that details the specific versions currently allowed. Where there are options or questions regarding the implementation of a specific Define-XML feature in the context of a regulatory submission, CDISC recommends consulting with the relevant authorities to ensure the implementation will be acceptable.

1.3 CDISC

The Clinical Data Interchange Standards Consortium (CDISC) is a nonprofit organization whose mission is to develop and support global, platform-independent data standards that enable information system interoperability to improve medical research and related areas of healthcare.

The US Food & Drug Administration (FDA) and the Pharmaceutical and Medical Devices Agency (PMDA) in Japan are long term-collaborators with CDISC. This collaboration has enabled greater standardization of the content and structure of clinical trials and nonclinical study data for regulatory submission. CDISC sponsors and members represent over 250 companies active in the research and development of regulated health-related products.

1.4 Operational Data Model (ODM)

The Define-XML standard is based on the CDISC Operational Data Model (ODM)-XML schema. ODM is a vendor-neutral, platform-independent format for the interchange and archiving of clinical study data. The model includes the clinical data along with its associated metadata, administrative data, reference data, and audit information. All of the information that needs to be shared among different software systems during setup, operation, analysis, and submission, or for long-term retention as part of an archive, is included in the model. ODM has been embraced by a broad range of clinical development organizations, and a number of vendors provide software applications and tools that use ODM. The current version of the ODM standard is available at http://www.cdisc.org/odm and the version of ODM supported for a specific version of the Define-XML is identified in the Define-XML schema.

One of the features of the ODM is a standardized mechanism for defining schema extensions to provide the functionality needed to support interchange requirements for specialized uses. To address the specific needs of data transmission in support of regulatory submissions, CDISC has developed the Define-XML model, which is implemented as a set of extensions to the base ODM schema. These extensions follow the guidelines for Vendor Extensions provided in the ODM specification and comply with the W3C XML Schema Version 1.0 specification. The XML schema files for the Define-XML standard are available online at http://www.cdisc.org/define-xml.

Although this document is intended to be understandable to readers with minimal technical knowledge of the ODM and XML, knowledge of this document alone is not a substitute for knowledge of the ODM nor is it sufficient to produce complete Define-XML documents. This document should be used in close concert with the current version of the ODM specification as well as current versions of the relevant data standards. The ODM specification package is available online at http://www.cdisc.org/odm. Numerous examples of XML fragments appear in this document. Many of these examples are provided as XML files and can be downloaded from the CDISC website (http://www.cdisc.org/define-xml).

1.5 Study Data Tabulation Model (SDTM)

The CDISC Study Data Tabulation Model (SDTM) defines a standard structure for case report form (CRF) data tabulations that are required to be submitted as part of a product application to the FDA. The CDISC SDTM is used to submit clinical trial and nonclinical study data for product applications across all therapeutic areas.

The current version of the SDTM standard is available at http://www.cdisc.org/sdtm. Define-XML v2.1 can be used to transmit metadata for clinical and nonclinical datasets that follow SDTM v1.2 and higher; this includes the SDTM

Implementation Guide (SDTMIG) v3.1.2 and higher, and the SEND Implementation Guide (SENDIG) v3.0 and higher.

1.5.1 SDTM Implementation Guide (SDTMIG)

The CDISC Study Data Tabulation Model Implementation Guide (SDTMIG) is based upon and must be used in close concert with the CDISC Study Data Tabulation Model (SDTM). The SDTM describes the conceptual model for representing study data for electronic interchange, and the SDTMIG provides specific domain models, assumptions, business rules, and examples for preparing standard clinical tabulation datasets based on the SDTM. The current version of the SDTMIG is available at https://www.cdisc.org/standards/foundational/sdtmig. Define-XML v2.1 can be used to transmit metadata for the SDTMIG v3.1.2 and higher.

1.5.2 SEND Implementation Guide (SENDIG)

The CDISC Standard for Exchange of Nonclinical Data Implementation Guide (SENDIG) is based upon and must be used in close concert with the CDISC Study Data Tabulation Model (SDTM). The SDTM describes the conceptual model for representing study data for electronic interchange, and the SENDIG provides specific domain models, assumptions, business rules, and examples for preparing standard nonclinical tabulation datasets based on the SDTM. The current version of the SENDIG standard is available at http://www.cdisc.org/send. Define-XML v2.1 can be used to transmit metadata for SENDIG v3.0 and higher.

1.6 Analysis Data Model (ADaM)

The CDISC Analysis Data Model (ADaM) defines standards for analysis datasets that are submitted as part of a product application to the FDA. In addition to defining fundamental principles that apply to all analysis datasets, ADaM defines standard structures that are appropriate for the majority of analysis datasets. In addition, the metadata to describe variable sources and derivations is of primary importance. Define-XML v2.1 can be used to transmit metadata for the ADaM Implementation Guide (ADaMIG) v1.0 and higher.

Because analysis datasets are developed to support specific analyses, ADaM has additional metadata that is not found in SDTM or SEND, notably analysis results metadata. The Analysis Results Metadata Standard extension to Define-XML will work with Define-XML 2.1. Appendix A explains how the original XML Schema files that were part of the Analysis Results Metadata Standard extension to Define-XML v2.0 can be updated to work with Define-XML v2.1.

The current version of the ADaM standard is available at https://www.cdisc.org/standards/foundational/adam.

2 Abbreviations and References

2.1 Definitions and Abbreviations

aCRF	Annotated Case Report Form
ADaM	Analysis Dataset Model
ADaMIG	ADaM Implementation Guide
BDS	Basic Data Structure (ADaM)
CDASH	Clinical Data Acquisition Standards Harmonization
CDISC	Clinical Data Interchange Standards Consortium
CRF	Case Report Form
CRT	Case Report Tabulation
eCTD	Electronic Common Technical Document
FDA	United States Food & Drug Administration
MedDRA	Medical Dictionary for Regulatory Activities
NCI	National Cancer Institute
OCCDS	ADaM Structure for Occurrence Data
ODM	Operational Data Model
OID	Object identifier
PDF	Portable Document Format
PDMA	Pharmaceuticals and Medical Devices Agency (Japan)
SAP	Statistical Analysis Plan
SDTM	Study Data Tabulation Model
SDTM-MSG	SDTM Metadata Submission Guidelines
SDTMIG	SDTM Implementation Guide
SDTMIG-MD	SDTMIG for Medical Devices
SEND	Standard for Exchange of Nonclinical Data
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XLink	XML Linking Language
XML	Extensible Markup Language
XPT	SAS Transport file
XSL	Extensible Stylesheet Language

2.2 References

Documents and resources referenced in the content of this Define-XML Specification may be accessed via the following links.

Analysis Data Model (ADaM) Version 2.1	http://www.cdisc.org/adam
ADaM Basic Data Structure (BDS) for Time-to-Event (TTE) Analyses v1.0	https://www.cdisc.org/standards/foundational/adam
ADaM Implementation Guide (ADaMIG) Version 1.1	https://www.cdisc.org/standards/foundational/adamig
Analysis Results Metadata (ARM) Version 1.0 for Define-XML Version 2.0	https://www.cdisc.org/standards/foundational/adam
CDISC Controlled Terminology	http://www.cancer.gov/cancertopics/cancerlibrary/terminologyresources/cdisc
CDISC Stylesheet Library	https://wiki.cdisc.org/display/PUB/Stylesheet+Library
CDISC website	http://www.cdisc.org
FDA Electronic Common Technical Document (eCTD) Guidance	https://www.fda.gov/drugs/developmentapprovalprocess/
FDA Regulatory Submission Guidance	https://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM072390.pdf
FDA Study Data Specifications	http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM312964.pdf
FDA Study Data Standards	http://www.fda.gov/forindustry/datastandards/studydatastandards/default.htm
IETF Trust, *Tags for Identifying Languages*	https://www.rfc-editor.org/pdfrfc/rfc5646.txt.pdf
National Cancer Institute (NCI) Enterprise Vocabulary Services	https://evs.nci.nih.gov
Operational Data Model (ODM) Version 1.3.2	http://www.cdisc.org/odm
SEND Implementation Guide (SENDIG) Version 3.1	http://www.cdisc.org/send
Study Data Tabulation Model (SDTM) Final Version 1.6	http://www.cdisc.org/sdtm
SDTM Implementation Guide (SDTMIG) Version 3.3	https://www.cdisc.org/standards/foundational/sdtmig
SDTM Metadata Submission Guidelines (SDTM-MSG) Version 1.0	https://www.cdisc.org/standards/foundational/sdtmmsg
W3C *Namespaces in XML 1.0*	https://www.w3.org/TR/xml-names/
W3C XML Schema Version 1.0	https://www.w3.org/XML/Schema
XML Schema Validation for Define.xml White Paper	http://www.cdisc.org/define-xml

3 Conformity and General Issues

This section supplements the CDISC Operational Data Model (ODM) v1.3.2 specification. All conformity requirements described in the ODM v1.3.2 specification are also applicable to Define-XML documents, which are based on the ODM.

3.1 File Conformity

The namespace URI for Version 2.1 of Define-XML is: http://www.cdisc.org/ns/def/v2.1. Throughout this document, the following conventions are used for namespaces:

- ODM elements and attributes are in the default namespace (i.e., they have no namespace prefix).
- Define-XML elements use the namespace prefix "def".
- Define-XML attributes use the namespace prefix "def" only if they appear within ODM elements.

Define-XML namespaces are defined in the ODM element.

Prefix	Usage	Allowable Variable (URL)	Description
xmlns	Required	http://www.cdisc.org/ns/odm/v1.3	Identifies the default namespace for this document
xmlns:def	Required	http://www.cdisc.org/ns/def/v2.1	XML namespace for Define-XML v2.1. Although "def:" is the suggested prefix for the Define-XML namespace, it should not be relied upon by the receiving application.
xmlns:xlink	Conditional Required when xlink:href is provided	http://www.w3.org/1999/xlink	XML namespace for XLink
xmlnl:xsi	Conditional Required when xlink:href is provided	http://www.w3.org/2001/XMLSchema-instance	XML Schema instance namespace. Required when xsi:schemaLocation is provided.

Note: These namespace prefixes are used throughout this document and are recommended as best practice to make it easier for users to understand and implement Define-XML, and to aid in the comparison of documents. In practice, other namespace prefixes can be used—as long as the Define-XML conforms to the rules of XML namespaces. Refer to the W3C for more information on this topic: https://www.w3.org/TR/xml-names/.

Any XML included in a Define-XML document that is not described in this specification is considered an extension.

Deprecated elements or attributes are not valid for use and are considered errors.

3.2 Extensions

The Define-XML schema permits vendor extensions, as defined in the ODM 1.3.2 specification, to the elements defined in this specification. These extensions may be CDISC-created extensions (e.g., Analysis Results Metadata extension) or vendor extensions.

Any XML not explicitly specified as part of Define-XML v2.1 is considered an extension. This includes ODM metadata not explicitly referenced in this specification. Extensions have no implied meaning with respect to the Define-XML standard; the sender and receiver must agree on a meaning themselves. That is, Define-XML documents that use extensions are not wrong, but rather the extensions may be ignored unless the sender and receiver have agreed otherwise. This also means that validators should flag ODM metadata not explicitly mentioned in the Define-XML specification with informational messages, and not with errors or warnings.

Requirements for vendor extensions to the Define-XML schema are:

- The vendor must supply an XML Schema fully describing the extended Define-XML format if using extended elements or attributes not already defined in the ODM namespace or Define-XML extension.
- Extended Define-XML documents should reference the proper extension schema.
- The extension may add new XML elements and attributes, as well as attribute values, but may not render any standard Define-XML elements or attributes obsolete. Vendor extensions may not be used for information that is normally expressed using other Define-XML elements or attributes.
- Elements and attributes from the ODM schema that are not a part of the Define-XML schema can be used as extensions, but no elements or attributes can be added to the ODM namespace.
- All extension elements and attributes not already defined in the ODM namespace must use a distinct XML namespace to ensure that there are no naming conflicts with other vendor extensions.
- The meaning of a Define-XML document must not be fundamentally changed by the addition of extensions.
- Removing all vendor extensions from an extended Define-XML document must result in a valid, meaningful, and accurate Define-XML document.
- Vendors should be able to produce Define-XML documents free of any vendor extensions upon request.

3.3 Define-XML Document Structure

Example 3.3.1 shows the structure of a valid Define-XML document. It illustrates a valid Define-XML document header and the set of elements that comprise this standard in the order in which they should appear. Note that this example includes elements that may be optional in some usage scenarios, such as arm:AnalysisResultDisplays and def:AnnotatedCRF.

Example 3.3.1 Define-XML Document Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="define2-1.xsl"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
     xmlns:def="http://www.cdisc.org/ns/def/v2.1"
     xmlns:xlink="http://www.w3.org/1999/xlink"
     xmlns:arm="http://www.cdisc.org/ns/arm/v1.0"
     FileOID="CDISC-Sample"
     ODMVersion="1.3.2"
     FileType="Snapshot"
     CreationDateTime="2018-11-15T11:01:00"
     def:Context="Submission">
    <Study>
        <GlobalVariables>
            <StudyName/>
            <StudyDescription/>
            <ProtocolName/>
        </GlobalVariables>
        <MetaDataVersion>
            <def:Standards>
                <def:Standard/>
            </def:Standards>
            <def:AnnotatedCRF>
                <def:DocumentRef/>
            </def:AnnotatedCRF>
            <def:SupplementalDoc>
                <def:DocumentRef/>
            </def:SupplementalDoc>
            <def:ValueListDef>
                <Description>
```

```
                    <TranslatedText/>
                </Description>
                <ItemRef>
                    <def:WhereClauseRef/>
                </ItemRef>
            </def:ValueListDef>
            <def:WhereClauseDef>
                <RangeCheck>
                    <CheckValue/>
                </RangeCheck>
            </def:WhereClauseDef>
            <ItemGroupDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <ItemRef/>
                <Alias/>
                <def:Class>
                    <def:SubClass/>
                </def:Class>
                <def:leaf>
                    <def:title/>
                </def:leaf>
            </ItemGroupDef>
            <ItemDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <CodeListRef/>
                <def:Origin>
                    <Description>
                        <TranslatedText/>
                    </Description>
                    <def:DocumentRef>
                        <def:PDFPageRef/>
                    </def:DocumentRef>
                </def:Origin>
                <def:ValueListRef/>
            </ItemDef>
            <CodeList>
                <Description>
                    <TranslatedText/>
                </Description>
                <EnumeratedItem>
                    <Alias/>
                    <Description>
                        <TranslatedText/>
                    </Description>
                </EnumeratedItem>
                <CodeListItem>
                    <Decode>
                        <TranslatedText/>
                        <Alias/>
                        <Description>
                            <TranslatedText/>
                        </Description>
                    </Decode>
                </CodeListItem>
                <ExternalCodeList>
                    <Alias/>
                </ExternalCodeList>
            </CodeList>
            <MethodDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <FormalExpression/>
                <def:DocumentRef>
                    <def:PDFPageRef/>
                </def:DocumentRef>
            </MethodDef>
            <def:CommentDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <def:DocumentRef>
                    <def:PDFPageRef/>
                </def:DocumentRef>
            </def:CommentDef>
            <def:leaf>
                <def:title/>
            </def:leaf>
         <arm:AnalysisResultDisplays/>
        </MetaDataVersion>
    </Study>
</ODM>
```

3.4 Ordering Elements and Attributes

3.4.1 Use of the OrderNumber Attribute

In several locations, this Define-XML specification uses an optional OrderNumber attribute to define the relative ordering of elements within a container. In the context of Define-XML, if the OrderNumber attribute is specified on an element, all elements of the same type (element name) within the same parent element must also specify an OrderNumber attribute. It is an error to mix elements with OrderNumber and without OrderNumber within a given parent element. If no OrderNumber attribute is specified in cases where order is important, the ordering is derived from the XML document order.

3.4.2 Other Order Considerations for Elements

This Define-XML specification does not require that ItemGroupDef elements related to the SDTM follow a particular order. However, for purposes of regulatory submissions, ItemGroupDef elements should follow the order recommended in the SDTM Metadata Submission Guidelines (MSG). Note that the SDTM-MSG indicates that the datasets should be displayed in the following Class order:

- TRIAL DESIGN
- SPECIAL PURPOSE – Subject Level
- INTERVENTIONS
- EVENTS
- FINDINGS
- FINDINGS ABOUT
- RELATIONSHIP
- STUDY REFERENCE

Within each class, datasets should be displayed in ascending alphabetic order by Name as maintained in the value of the ItemGroupDef Name attribute.

A standard order of display has not been established for regulatory submissions of ADaM datasets.

3.5 Defs and Refs

This document sometimes references elements as "Defs" and "Refs". In Define-XML, an element whose name ends with "Def" is the declaration of an object instance. An element whose name ends with "Ref" is a reference to that object from some other entity.

For example, in this specification variables are declared using elements named ItemDef. However, to indicate that an ItemGroup includes a particular item instance, an ItemRef element is used to reference the appropriate ItemDef object.

3.5.1 OIDs

Attributes whose names end with "OID" (sometimes called *OIDs)* are used to uniquely identify different metadata objects. This Define-XML specification does not mandate a format for OIDs. OIDs are only intended as a mechanism for unambiguously linking between a definition of an object and references to it. The examples in this document use prefixes at the start of OIDs to indicate the object type. However, this is not required; it is equally valid to use randomly generated identifiers.

Example 4.5.2.4 illustrates the concepts of linking between a definition of an object and references to it.

OIDs play a critical role in presenting Define-XML as one interconnected document containing a web of multiple parts that are linked together to form a single set of metadata. OIDs provide these links, allowing one to state the variables of a dataset and yet define those variables outside of the dataset context. They permit naming a list of allowable values for a variable, and yet defining those lists of values outside of the variable context. Because definitions can exist outside of any context, any one definition can be referenced from multiple contexts. For example, the variable STUDYID is found in almost every SDTM dataset. By defining STUDYID outside of any dataset context, we can define it only once and reference the definition from every dataset that contains the variable. In this sense, the STUDYID definition is being *shared* or *reused* by all the datasets.

The value of an OID attribute has no intrinsic meaning. Although some may find convenience in applying a predefined convention for assigning values to aid readability (as illustrated throughout this document), no meaning should be attached to the OID value. The significance of an OID value is simply that it is used to reference an object defined elsewhere, and no additional semantics are implied. For example, the definition of a variable referenced in an ItemRef element contained within an ItemGroupDef element can be found in the one and only ItemDef element (outside of ItemGroupDef) whose OID attribute value matches the value of the ItemRef ItemOID attribute. Similar associations are made between variables and their corresponding codelist definitions, variables and their corresponding valuelist definitions, and in other sections of the document. When studying the examples throughout this document, take note of these OID value pairings.

3.6 Validation of a Define-XML v2.1 Document

A valid Define-XML document must:

- properly reference versions of the CDISC standards,
- be well formed and conform to the XML schemas, and
- meet all of the requirements documented in this specification.

Once a Define-XML document is valid according to the schema, validation software should consider all other Define-XML requirements in the specification. These include rules about conditionally required components or other business rules in this document. The Define-XML schema can only enforce some of the standard; this additional level of validation is required to determine if a Define-XML document is compliant with Define-XML v2.1. See the XML Schema Validation for Define.xml White Paper (http://www.cdisc.org/define-xml) for additional information.

The correct ordering of elements within a document is an absolute requirement for the document to be valid with respect to the Define-XML schema. The use of an XML-Schema definition and a validating parser environment makes detection of improperly ordered content fairly straightforward. In the absence of such mechanisms, care should be extended to following the order specified by the documentation for all extension content.

Note: XML is case-sensitive, and case sensitivity plays a role in creating a valid Define-XML document. For example, <def:Class Name="findings" /> is not valid, but <def:Class Name="FINDINGS" /> is valid because it uses the capitalized version of "FINDINGS" included in this specification.

4 General Specifications for Define-XML

The purpose of Define-XML is to support the interchange of dataset metadata for clinical research applications in a machine-readable format. An important use case for Define-XML is submitting clinical trials data in CDISC SDTM, SEND, or ADaM format to regulatory authorities. The key metadata components to support submissions are:

- References to versions of content standards
- Dataset definitions
- Dataset variable definitions
- Controlled Terminology definitions
- Value list definitions
- Links to supporting documents (e.g., reviewers' guides)
- Computational method definitions
- Comment definitions

4.1 Standards References

Each Define-XML document will provide references to 1 or more published content standards. For regulatory data submissions, CDISC content standards are typically used. Regulatory authority technical conformance guides typically list the specific standards permitted in data submissions and provide the support dates for specific standards versions. Although in many cases the metadata described in a Define-XML document will follow a single version of a content standard, there are cases where a subset of the datasets within a data submission may follow a newer version. Consultation with the relevant regulatory authorities prior to creating a submission with mixed standards versions is strongly recommended.

The def:Standards element (see Section 5.3.6 def:Standards Element) within MetaDataVersion lists the content standards used within a Define-XML document. The def:Standards element must contain 1 or more def:Standard elements. Each def:Standard element specifies the standard Name, Type, and Status.

For a submission that includes tabulation domains from two or more versions of the SDTMIG or SENDIG, the def:Standards element will contain multiple def:Standard elements; 1 for each standard version associated with submitted domains. In this case, the definitions for each dataset will include a def:StandardOID attribute that references the appropriate version of the SDTMIG or SENDIG.

Use of attribute def:IsNonStandard for SDTM and SEND

A dataset is considered non-standard if it is either a sponsor-defined custom domain or a domain based on an unpublished draft of a CDISC dataset. For non-standard domain definitions, the ItemGroupDef for the dataset must include a def:IsNonStandard attribute.

In most cases, all variables within a dataset will be from the same content standard version. If a dataset variable is not part of the def:Standard referenced by the dataset, it is considered non-standard for regulatory submission purposes; even in the case where the variable is defined in a different version of the same standard, in the context of a regulatory submission the ItemRef element for the variable will include the attribute def:IsNonStandard. The use of non-standard variables, whether based on an unpublished draft of a CDISC dataset or part of a sponsor-defined custom domain, should be discussed with the relevant regulatory authorities prior to use within a submission.

Use of attribute def:IsNonStandard for ADaM

An analysis dataset included in the ADaM folder for regulatory submission is considered non-standard if it is a non-ADaM analysis dataset as defined in Section 1.6 of ADaMIG V1.1. For this type of dataset, the ItemGroupDef must include the def:IsNonStandard attribute. Because the non-ADaM analysis datasets do not belong to any of the valid ADaM dataset classes, the def:Class element will not be used for this type of dataset. On the variable level, the def:IsNonStandard has no meaning for ADaM, so it should generally not be used for ADaM variable metadata.

Note: The def:Standards and def:Standard elements replace the def:StandardName and def:StandardVersion attributes used in Define-XML Version 2.0.

4.1.1 Standards Considerations

The following table lists the content standard types and their associated allowable names. These terms will be managed and published with CDISC Controlled Terminology.

Type	Description	Allowable Names	Publishing Set
IG	Implementation Guide based on a given CDISC standard model **The Model is implicit in the name of the IG.** It is expected that the CDISC Standards Repository (SHARE) will describe the base model for each IG.	• SDTMIG • SDTMIG-MD • SDTMIG-AP • SDTMIG-PGx • SENDIG • SENDIG-DART • ADaMIG	
CT	Controlled Terminology	CDISC/NCI	• CDASH • SDTM • ADaM • SEND

The following table lists examples of the available versions of standards for 2 allowable standard names. For a complete list of available versions, refer to http://cdisc.org/standards.

Name	Allowable Versions
SDTMIG	3.1.2 3.2 3.3
SDTMIG-MD	1.0

The value of the Status attribute is expected to be managed and published with CDISC Controlled Terminology.

Attribute	Description	Allowable Values	Default
Status	Development or publishing status of the standard	Draft Provisional Final	Final

It is important to check the CDISC web page (http://cdisc.org/standards) for the relevant content standard to ensure that the value of the Status attribute is accurate.

4.1.2 Examples of Standards Metadata in Define-XML

4.1.2.1 Single SDTMIG and Controlled Terminology Reference

In the simplest case, 1 version of the SDTMIG standard and 1 version of a CDISC/NCI Controlled Terminology publishing set is used for all datasets within a submission.

- The MetaDataVersion includes a def:CommentOID with the value COM.MDV that identifies a def:CommentDef element that includes a reference to the study standards usage section of the Study Data Reviewers Guide document. (**Note**: The def:CommentDef element is not shown here.)
- The def:Standards element has no attributes but serves as a container for def:Standard elements that list the CDISC standards referenced in the Define-XML document. The def:Standards element appears immediately following the MetaDataVersion element. Although the def:Standards element is required only

when the ODM def:Context attribute value is Submission, it may be useful in any Define-XML that represents CDISC metadata.

- The value of the Name attribute should appear on Define-XML stylesheet displays. Valid values for the Name will be published as extensible CDISC Controlled Terminology.
- The value of the Type attribute identifies whether the standard is an implementation guide or controlled terminology. Valid values for the Type will be published as extensible CDISC Controlled Terminology.
- The PublishingSet attribute is used only for controlled terminology standards. Valid values for PublishingSet will be published as CDISC Controlled Terminology.
- The Version attribute value is a text string. For CDISC foundational standards, the Version value must exactly match the version number of the standard used to that on the CDISC web page for that standard. For CDISC Controlled Terminology, the Version will identify the publication date using ISO 8601-formatted date string.
- The Status attribute identifies the publishing status for the standard version. Valid values for Status will be published as CDISC Controlled Terminology.

Example 4.1.2.1.1 Simple Standard Reference

```
 <MetaDataVersion OID="MDV.001"  Name="Study CDISC01, Data Definitions, V-001"
Description="Study CDISC01, Data Definitions, First release"
def:DefineVersion="2.1" def:CommentOID="COM.MDV">
     <def:Standards>
        <def:Standard OID="STD.SDTMIG-3.2"
                      Name="SDTMIG"
                      Type="IG"
                      Version="3.2"
                      Status="Final"/>
        <def:Standard OID="STD.CT-20141219"
                      Name="CDISC/NCI"
                      Type="CT"
                      PublishingSet="SDTM"
                      Version="2014-12-19"
                      Status="Final"/>
     </def:Standards>
```

4.1.2.2 Multiple Standard Version References

Example 4.1.2.2.1 illustrates a case where the submission includes a domain that is defined in the SDTMIG for Medical Devices (SDTMIG-MD).

- The def:Standard element with the OID value STD.SDTMIG-3.1.3 identifies the base SDTMIG version; the def:Standard element with the OID value STD.SDTMIG-MD-1.0 identifies the SDTMIG-MD v1.0. Because the SDTMIG-MD was published as a provisional standard, the Status attribute has the value "Provisional".
- It is important to check the SDTM Standards page on the CDISC website to ensure that value of the Status attribute is accurate at the time the Define-XML is submitted to a regulatory authority.
- Each ItemGroupDef includes a def:StandardOID attribute.
- In the ItemGroupDef element for the Demographics domain, the value for the def:StandardOID attribute is STD.SDTMIG-3.1.3.
- In the ItemGroupDef element for the Device Events domain, the value for the def:StandardOID attribute is STD.SDTMIG-MD-1.0.

Example 4.1.2.2.1 References to Multiple SDTM Standards

```
   <MetaDataVersion OID="MDV.CDISC01.SDTMIG.3.1.3.SDTM.1.3" Name="Study CDISC01,
Data Definitions"
      Description="Study CDISC01, Data Definitions" def:DefineVersion="2.1.0"
def:CommentOID="COM.MDV">
      <def:Standards>
            <def:Standard OID="STD.SDTMIG-3.1.3" Name="SDTMIG" Type="IG"
Version="3.1.3" Status="Final"/>
            <def:Standard OID="STD.SDTMIG-MD-1.0" Name="SDTMIG-MD" Type="IG"
Version="1.0" Status="Provisional"/>
      </def:Standards>
      <ItemGroupDef OID="IG.DM" Domain="DM" Name="DM" Repeating="No"
Purpose="Tabulation" def:StandardOID="STD.SDTMIG-3.1.3"
            def:Structure="One record per subject">
            <Description><TranslatedText
xml:lang="en">Demography</TranslatedText></Description>
            <ItemRef ItemOID="IT.DM.STUDYID" OrderNumber="1" Mandatory="Yes"/>
             ...
            <def:Class Name="SPECIAL PURPOSE"/>
       </ItemGroupDef>
     <ItemGroupDef OID="IG.DE"  Domain="DE" Name="DE" Repeating="Yes"
IsReferenceData="No" SASDatasetName="DE"
      Purpose="Tabulation"      def:Structure="One record per event per
device"
      def:ArchiveLocationID="LF.DE"      def:StandardOID="STD.SDTMIG-MD-1.0">
      <Description>
            <TranslatedText xml:lang="en">Device Events</TranslatedText>
      </Description>
      ...
      <ItemRef ItemOID="IT.DE.DETERM" OrderNumber="7" Mandatory="Yes"/>
.      ..
      <def:leaf ID="LF.DE" xlink:href="de.xpt">
      <def:title>de.xpt</def:title>
      </def:leaf>
      </ItemGroupDef>
```

4.1.2.3 Multiple SDTMIG Versions

In this example, the Opthalmic Examinations dataset follows SDTMIG 3.3. The Adverse Events dataset follows SDTMIG 3.2, but adds the variable with ItemOID="IT.FOCID" identifier as a non-standard variable.

- The def:Context attribute on the ODM root element has the value Other to indicate that this Define-XML document is not intended to be used for a regulatory submission.
- In the ItemGroupDef element for the OE dataset, the value for the def:StandardOID attribute is STD.SDTMIG-3.3 and the value for the Status attribute is DRAFT.
- In the ItemGroupDef element for the AE dataset, the value for the def:StandardOID attribute is STD.SDTMIG-3.2. The value for the Status attribute is Final because this is a published version of the SDTMIG.
- In the ItemRef element for the FOCID variable, the optional def:IsNonStandard attribute is used to indicate that this is a non-standard SDTMIG variable. When the def:IsNonStandard attribute is provided, the value should always be Yes.

Example 4.1.2.3.1 Non-Standard Variables

```
<ODM
  xmlns="http://www.cdisc.org/ns/odm/v1.3"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:def="http://www.cdisc.org/ns/def/v2.1"
  ODMVersion="1.3.3"
  FileOID="www.cdisc.org.Studycdisc01-Define-XML_2.1.0"
  FileType="Snapshot"
  CreationDateTime="2018-04-30T14:25:44"
  Originator="CDISC Data Exchange Standards Team" def:Context="Other">
  <MetaDataVersion OID="MDV.CDISC01.SDTMIG.3.1.2.SDTM.1.2" Name="Study CDISC01,
Data Definitions"
    Description="Study CDISC01, Data Definitions"
def:DefineVersion="2.1.0"def:CommentOID="COM.MDV">
    <def:Standards>
        <def:Standard OID="STD.SDTMIG-3.3" Name="SDTMIG" Type="IG" Version="3.3"
Status="Final" />
        <def:Standard OID="STD.SDTMIG-3.2" Name="SDTMIG" Type="IG" Version="3.2"
Status="Final"/>
    </def:Standards>
    <ItemGroupDef OID="IG.AE" Domain="AE" Name="AE" Repeating="Yes"
IsReferenceData="No" SASDatasetName="AE"
        Purpose="Tabulation" def:Structure="One record per adverse event per
subject"
        def:ArchiveLocationID="LF.AE" def:StandardOID="STD.SDTMIG-3.2">
        <Description>
            <TranslatedText xml:lang="en">Adverse Events</TranslatedText>
        </Description>
              ...
        <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
        <ItemRef ItemOID="IT.AE.DOMAIN" OrderNumber="2" Mandatory="Yes"/>
        <ItemRef ItemOID="IT.USUBJID" OrderNumber="3" Mandatory="Yes"
KeySequence="2" MethodOID="MT.USUBJID"/>
        <ItemRef ItemOID="IT.FOCID" OrderNumber="4" Mandatory="No"
def:IsNonStandard="Yes"/>
              ...
        <def:Class Name="EVENTS"/>
        <def:leaf ID="LF.AE" xlink:href="ae.xpt">
             <def:title>AE.xpt</def:title>
        </def:leaf>
    </ItemGroupDef>
    <ItemGroupDef OID="IG.OE" Domain="OE" Name="OE" Repeating="Yes"
IsReferenceData="No" SASDatasetName="OE"
        Purpose="Tabulation" def:Structure="One record per ophthalmic finding per
method per location per time point per visit per subject”
        def:ArchiveLocationID="LF.OE" def:StandardOID="STD.SDTMIG-3.3">
        <Description>
           <TranslatedText xml:lang="en">Ophthalmic Examinations</TranslatedText>
        </Description>
        <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
        <ItemRef ItemOID="IT.AE.DOMAIN" OrderNumber="2" Mandatory="Yes"/>
        <ItemRef ItemOID="IT.USUBJID" OrderNumber="3" Mandatory="Yes"
KeySequence="2" MethodOID="MT.USUBJID"/>
        <ItemRef ItemOID="IT.FOCID" OrderNumber="4" Mandatory="No"/>
        …
        <def:Class Name="FINDINGS"/>
        <def:leaf ID="LF.OE" xlink:href="oe.xpt">
```

```
            <def:title>oe.xpt</def:title>
        </def:leaf>
    </ItemGroupDef>
    <ItemDef OID="IT.FOCID" Name="FOCID" DataType="text" Length="2"
SASFieldName="FOCID">
        <Description>
            <TranslatedText xml:lang="en"> Focus of Study-Specific
Interest</TranslatedText>
        </Description>
        <CodeListRef CodeListOID="CL.FOCID"/>
        <def:Origin Type="Collected" Source="Investigator">
          <def:DocumentRef leafID="LF.blankcrf">
            <def:PDFPageRef PageRefs="35" Type="PhysicalRef"/>
          </def:DocumentRef>
        </def:Origin>
     </ItemDef>
```

4.1.2.4 ADaMIG Reference

This example illustrates a case where the Define-XML document for an ADaM submission references an ADaMIG version and 2 CDISC Controlled Terminology publishing sets.

- The def:Standards element includes 3 def:Standard elements.
- The def:Standard element with the OID value STD.ADaMIG-1.1 identifies ADaMIG v1.1.
- The def:Standard element with the OID value CT.SDTMCT-Q12018 identifies the SDTM publishing set of the CDISC Controlled Terminology with the Version identifier 2018-03-30.
- The def:Standard element with the OID value CT.ADaMCT-Q32017 identifies the ADaM publishing set of the CDISC Controlled Terminology with the Version identifier 2017-09-29.
- The ItemGroupDef element with the OID value IG.ADSL includes the def:StandardOID value STD.ADaMIG-1.1 to indicate that it references the def:Standard element with that OID.
- The CodeList element with the OID value CL.C66781.AGEU includes a def:StandardOID attribute with the value CT.SDTMCT-Q12018 to indicate that it follows the controlled terminology publication identified by the def:Standard element with the matching OID value.
- The CodeList element with the OID value CL.C81224.DTYPE includes a def:StandardOID attribute with the value CT.ADAMCT.Q32017 to indicate that it follows the controlled terminology publication identified by the def:Standard element with the matching OID value

Note: Each ItemGroupDef element used to provide metadata for a dataset that follows a published CDISC standard includes a def:StandardOID attribute that matches the OID for a def:Standard element in the def:Standards element in the Define-XML document. Each CodeList element used to provide a code list that is taken from a CDISC controlled terminology publication must include a def:StandardOID attribute that matches the OID for a def:Standard element in the def:Standards container element in the Define-XML document.

Example 4.1.2.4.1 Use of def:Standard for ADaM

```
<MetaDataVersion OID="MDV.CDISC01.ADaMIG.1.1"
      Name="Study CDISC-Sample, Data Definitions"
      Description="Study CDISC-Sample, Data Definitions"
      def:DefineVersion="2.1.0">
      <def:Standards>
        <def:Standard OID="STD.ADaMIG-1.1" Name="ADaMIG" Type="IG" Version="1.1"
Status="Final"/>
```

```
        <def:Standard OID="CT.SDTMCT-Q12018" Name="SDTM Terminology" Type="CT"
PublishingSet="SDTM" Version="2018-03-30" Status="Final"/>
        <def:Standard OID="CT.ADaMCT-Q32017" Name="ADaM Terminology" Type="CT"
PublishingSet="ADaM" Version="2017-09-29" Status="Final"/>
      </def:Standards>
             <ItemGroupDef OID="IG.ADSL" Name="ADSL" SASDatasetName="ADSL"
Repeating="No" Purpose="Analysis" def:Structure="one record per subject"
        def:CommentOID="COM.ADSL" def:ArchiveLocationID="LF.ADSL"
def:StandardOID="STD.ADaMIG-1.1">
        <Description>
          <TranslatedText xml:lang="en">Subject-Level Analysis</TranslatedText>
        </Description>
        <ItemRef ItemOID="IT.ADSL.STUDYID" OrderNumber="1" Mandatory="No"/>
              ...
        <def:Class Name="SUBJECT LEVEL ANALYSIS DATASET"/>
        <def:leaf ID="LF.ADSL" xlink:href="adsl.xpt">
             <def:title>adsl.xpt</def:title>
        </def:leaf>
    </ItemGroupDef>


    ...

       <CodeList OID="CL.C66781.AGEU" Name="Age Unit" DataType="text"
def:StandardOID="CT.SDTMCT-Q12018">
        <EnumeratedItem CodedValue="YEARS">
          <Alias Name="C29848" Context="nci:ExtCodeID"/>
        </EnumeratedItem>
        <Alias Name="C66781" Context="nci:ExtCodeID"/>
      </CodeList>
    ...
    <CodeList OID="CL.C81224.DTYPE" Name="Derivation Type" DataType="text"
def:StandardOID="CT.ADaMCT-Q32017">
        <CodeListItem CodedValue="LOCF">
          <Decode>
            <TranslatedText xml:lang="en">Last Observation Carried
Forward</TranslatedText>
          </Decode>
          <Alias Name="C81198" Context="nci:ExtCodeID"/>
        </CodeListItem>
        <Alias Name="C81224" Context="nci:ExtCodeID"/>
      </CodeList>
```

4.2 Dataset Definitions

The CDISC SDTM, SEND, and ADaM study datasets are modeled as tables where the columns represent variables and the rows represent observed or derived values of those variables. Dataset metadata is represented in Define-XML as an ItemGroupDef element (see Section 5.3.11, ItemGroupDef Element).

Dataset definitions include references to the definitions of dataset variables and, when used for regulatory submissions, a reference to the file containing the dataset data content. In Define-XML, the references to dataset variables are specified using the ItemRef element (see Section 5.3.9.2, ItemRef Element) and the reference to the dataset file is provided by a def:leaf element (see Section 5.3.16, def:leaf Element). Sponsors may provide SDTM comments or ADaM documentation references at the dataset level. Comments are provided in Define-XML using the def:CommentDef element (see Section 5.3.15, def:CommentDef Element).

The SDTM and ADaM standards use Class as a way to identify each dataset in relation to its respective model. Beginning with Version 2.1, the Class concept has been expanded to include a SubClass. The initial use cases are to identify Adverse Events Analysis datasets within the ADaM OCCURRENCE DATA STRUCTURE class or Time to Event datasets within the ADaM Basic Data Structure class. It is likely that over time SubClass will be used in additional cases for ADaM and SDTM datasets. Users should consult the implementation guide for the relevant standard and CDISC Controlled Terminology for more specific information about use of SubClass. Each of the examples of dataset definitions in this section includes Class metadata.

4.2.1 Examples of Dataset Metadata in Define-XML

4.2.1.1 Non-repeating SDTM Dataset

This is an example of an ItemGroupDef element that represents the SDTM Demographics domain dataset metadata.

- The SDTM DM domain has only 1 record for each subject, so the Repeating attribute is set to No.
- The def:StandardOID attribute has the value STD.SDTMIG-3.2, so a def:standard element with an OID attribute set to this value must be included in the def:Standards element in this Define-XML document.
- This ItemGroupDef contains 1 ItemRef element for each variable in the DM domain. For variables that are designated as Required in the SDTMIG referenced by the def:StandardOID attribute, the Mandatory attribute should be set to Yes. For variables that are designated as Expected or Permissible in the implementation guide, the Mandatory attribute should normally be set to No. If, however, the sponsor chooses to designate a variable as Required that the SDTMIG designates as Permissible or Expected, then the Mandatory attribute should be set to Yes. In this example, AGE (designated as Expected by the SDTMIG) and ETHNIC (designated as Permissible by the SDTMIG) are set to Mandatory because the sponsor decided to designate both as Required.
- The Domain attribute of the ItemGroupDef element is required as specified by the SDTM Metadata Submission Guidelines (SDTM-MSG).

Example 4.2.1.1.1 Non-repeating Dataset

```
 <!-- Dataset Definition (DM) -->
<ItemGroupDef OID="IG.DM"
    Domain="DM"
    Name="DM"
    Repeating="No"
    IsReferenceData="No"
    SASDatasetName="DM"
    Purpose="Tabulation"
    def:Structure="One record per subject"
    def:CommentOID="COM.DOMAIN.DM"
    def:ArchiveLocationID="LF.DM"
    def:StandardOID="STD.SDTMIG-3.2">
    <Description>
        <TranslatedText xml:lang="en">Demographics</TranslatedText>
    </Description>
    <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
    <ItemRef ItemOID="IT.DM.DOMAIN" OrderNumber="2" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.USUBJID" OrderNumber="3" Mandatory="Yes" KeySequence="2”
MethodOID="MT.USUBJID"/>
    <ItemRef ItemOID="IT.DM.SUBJID" OrderNumber="4" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.DM.RFSTDTC" OrderNumber="5" Mandatory="No"
MethodOID="MT.RFSTDTC"/>
    <ItemRef ItemOID="IT.DM.RFENDTC" OrderNumber="6" Mandatory="No"
MethodOID="MT.RFENDTC"/>
    <ItemRef ItemOID="IT.DM.SITEID" OrderNumber="7" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.DM.BRTHDTC" OrderNumber="8" Mandatory="No"/>
    <ItemRef ItemOID="IT.DM.AGE" OrderNumber="9" Mandatory="Yes"
MethodOID="MT.AGE"/>
```

```
    <ItemRef ItemOID="IT.DM.AGEU" OrderNumber="10" Mandatory="No"/>
    <ItemRef ItemOID="IT.DM.SEX" OrderNumber="11" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.DM.RACE" OrderNumber="12" Mandatory="No"/>
    <ItemRef ItemOID="IT.DM.ETHNIC" OrderNumber="13" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.DM.ARMCD" OrderNumber="14" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.DM.ARM" OrderNumber="15" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.DM.COUNTRY" OrderNumber="16" Mandatory="Yes"/>
    <def:Class Name="SPECIAL PURPOSE"/>
    <def:leaf ID="LF.DM" xlink:href="dm.xpt">
       <def:title>dm.xpt</def:title>
    </def:leaf>
</ItemGroupDef>
```

4.2.1.2 SDTM Reference Dataset

Example 4.2.1.2.1 illustrates the use of the ItemGroupDef element to provide metadata for the SDTM Trial Elements domain dataset.

- The SDTM TE domain contains no subject-level data, so the IsReferenceData attribute is included and is set to Yes.
- The Name attribute for the def:Class child element for the Trial Events datasets is TRIAL DESIGN.

Example 4.2.1.2.1 Reference Dataset/Trial Elements Domain Dataset

```
<!-- Dataset Definition (TE) -->
<ItemGroupDef OID="IG.TE"
    Domain="TE"
    Name="TE"
    Repeating="No"
    IsReferenceData="Yes"
    SASDatasetName="TE"
    Purpose="Tabulation"
    def:Structure="One record per planned Element"
    def:ArchiveLocationID="LF.TE"
    def:StandardOID="STD.SDTM-3.2">
    <Description>
        <TranslatedText xml:lang="en">Trial Elements</TranslatedText>
    </Description>
    <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
    <ItemRef ItemOID="IT.TE.DOMAIN" OrderNumber="2" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.TE.ETCD" OrderNumber="3" Mandatory="Yes"
KeySequence="2"/>
    <ItemRef ItemOID="IT.TE.ELEMENT" OrderNumber="4" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.TE.TESTRL" OrderNumber="5" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.TE.TEDUR" OrderNumber="6" Mandatory="No"/>
    <def:Class Name="TRIAL DESIGN"/>
    <def:leaf ID="LF.TE" xlink:href="te.xpt">
        <def:title>te.xpt</def:title>
    </def:leaf>
</ItemGroupDef>
```

4.2.1.3 SDTM Split Dataset

Example 4.2.1.3.1 illustrates the construction of ItemGroupDef elements to provide metadata for 2 split datasets for the SDTM Questionnaire domain.

- The value of the Domain element for both ItemGroupDef elements is "QS". The Name and SASDatasetName attributes contain the split dataset names. Each ItemGroupDef element includes a Description child element with a label for the split dataset, and an Alias child element with the Context attribute set to "DomainDescription" to indicate that its Name attribute provides a label for the full domain dataset.
- Each ItemGroupDef element includes attributes for the def:ArchiveLocationID and the def:CommentOID. These reference a def:leaf child element describing the location of the dataset file and a def:Comment element, respectively.
- The datasets follow the naming recommendations specified in Section 4.1.1.7 of the SDTMIG 3.2 and Section 4.1.7 of the SDTMIG 3.3. Not all ItemRef elements are included in this example.

Note: Because split-domain datasets may be combined by the data recipient, the variables belonging to each split must be defined in a consistent manner. ItemRef definitions may be reused, as illustrated by the use of the same ItemOID values across the 2 datasets, but if the variables differ in any of their metadata (e.g. Value Level metadata) across the split, separate ItemRef elements corresponding to distinct ItemDef elements are needed.

Example 4.2.1.3.1 Split SDTM Dataset Definition

```
<!-- Dataset Definition (QSCG) -->
<ItemGroupDef OID="IG.QSCG"
    Domain="QS"
    Name="QSCG"
    Repeating="Yes"
    IsReferenceData="No"
    SASDatasetName="QSCG"
    Purpose="Tabulation"
    def:Structure="One record per questionnaire per question per visit per
subject"
    def:CommentOID="COM.DOMAIN.QS"
    def:ArchiveLocationID="LF.QSCG"
    def:StandardOID="STD.SDTM-3.2">
    <Description>
        <TranslatedText xml:lang="en">Questionnaire-QSCG</TranslatedText>
    </Description>
    <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
    …
    <ItemRef ItemOID="IT.QS.QSDY" OrderNumber="17" Mandatory="No"
MethodOID="MT.QSDY"/>
    <Alias Context="DomainDescription" Name="Questionnaires"/>
    <def:Class Name="FINDINGS"/>
    <def:leaf ID="LF.QSCG" xlink:href="qscg.xpt">
    <def:title>qscg.xpt</def:title>
    </def:leaf>
</ItemGroupDef>
<!-- Dataset Definition (QSCS) -->
<ItemGroupDef OID="IG.QSCS"
    Domain="QS"
    Name="QSCS"
    Repeating="Yes"
    IsReferenceData="No"
    SASDatasetName="QSCS"
    Purpose="Tabulation"
    def:Structure="One record per questionnaire per question per visit per
subject"
    def:CommentOID="COM.DOMAIN.QS"
    def:ArchiveLocationID="LF.QSCS">
    <Description>
        <TranslatedText xml:lang="en">Questionnaire-QSCS</TranslatedText>
    </Description>
    <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
    …
    <ItemRef ItemOID="IT.QS.QSDY" OrderNumber="17" Mandatory="No"
MethodOID="MT.QSDY"/>
    <ItemRef ItemOID="IT.QS.QSEVLINT" OrderNumber="18" Mandatory="No"/>
    <Alias Context="DomainDescription" Name="Questionnaires"/>
    <def:Class Name="FINDINGS"/>
    <def:leaf ID="LF.QSCS" xlink:href="qscs.xpt">
    <def:title>qscs.xpt</def:title>
    </def:leaf>
</ItemGroupDef>
<!-- Dataset Definition (QSMM) -->
<ItemGroupDef OID="IG.QSMM" Domain="QS"

…

<!-- Comment Definition: Long Comment, included in a PDF file -->
<def:CommentDef OID="COM.DOMAIN.QS">
    <Description>
        <TranslatedText xml:lang="en"> QS is submitted as a split dataset. The
split was done based on QSCAT as QSCG (CLINICAL GLOBAL IMPRESSIONS), QSCS
(CORNELL SCALE FOR DEPRESSION IN DEMENTIA) and QSMM (MINI MENTAL STATE
EXAMINATION). See
        additional documentation in the Reviewer's Guide, Split Datasets Section.
        </TranslatedText>
    </Description>
    <def:DocumentRef leafID="LF.ReviewersGuide"/>
</def:CommentDef>
```

4.2.1.4 ADaM Time to Event Dataset

The dataset defined in this example is named ADPFS. It was built according to the ADaM Basic Data Structure (BDS) for Time-to-Event (TTE) Analyses.

- Because time-to-event analysis is a specific application of the Basic Data Structure (BDS), the Name attribute of the def:Class element is BASIC DATA STRUCTURE and the Name attribute for the def:SubClass child element of the def:Class element is TIME-TO-EVENT.
- The use of the def:SubClass child element for specific ADaM datasets facilitates validation of these datasets, because specific variables must be included in a BDS dataset when it includes time-to-event parameters.

Example 4.2.1.4.1 ADaM Time to Event Dataset Definition

```
<ItemGroupDef OID="IG.ADPFS"
        Name="ADPFS"
        Repeating="Yes"
        IsReferenceData="No"
        SASDatasetName="ADPFS"
        Purpose="Analysis"
        def:Structure="One record per subject per parameter"
        def:ArchiveLocationID="LF.ADPFS">
        <Description>
          <TranslatedText xml:lang="en">Time to Progression Free
Survival</TranslatedText>
        </Description>
        <ItemRef ItemOID="IT.ADPFS.STUDYID" OrderNumber="1" Mandatory="No"
KeySequence="1"/>
        <ItemRef ItemOID="IT.ADPFS.USUBJID" OrderNumber="2" Mandatory="No"
KeySequence="2"/>
 …
        <def:Class Name="BASIC DATA STRUCTURE">
          <def:SubClass Name="TIME-TO-EVENT"/>
        </def:Class>
        <def:leaf ID="LF.ADPFS" xlink:href="adpfs.xpt">
          <def:title>adpfs.xpt</def:title>
        </def:leaf>
      </ItemGroupDef>
```

4.2.1.5 ADaM Occurrence Dataset

The dataset defined in this example is named ADAE.

- The Name attribute for the def:Class child-element is OCCURRENCE DATA STRUCTURE. The Name attribute for the def:SubClass child element of the def:Class element is ADVERSE EVENT.
- Except in cases where def:SubClass is defined by the ADaM team, it is sufficient to specify def:Class for other ADaM datasets.

Example 4.2.1.5.1 ADaM Occurrence Dataset Definition

```
<!-- Dataset Definition (ADAE) -->
      <ItemGroupDef OID="IG.ADAE"
        Name="ADAE"
        Repeating="Yes"
        IsReferenceData="No"
        SASDatasetName="ADAE"
        Purpose="Analysis"
        def:Structure="One record per subject per event"
        def:ArchiveLocationID="LF.ADAE">
        <Description>
          <TranslatedText xml:lang="en">Adverse Event Analysis
Dataset</TranslatedText>
        </Description>
        <ItemRef ItemOID="IT.ADAE.STUDYID" OrderNumber="1" Mandatory="No"
KeySequence="1"/>
        <ItemRef ItemOID="IT.ADAE.USUBJID" OrderNumber="2" Mandatory="No"
KeySequence="2"/>
 …
        <def:Class Name="OCCURRENCE DATA STRUCTURE">
          <def:SubClass Name="ADVERSE EVENT"/>
        </def:Class>
        <def:leaf ID="LF.ADAE" xlink:href="adae.xpt">
          <def:title>adae.xpt</def:title>
        </def:leaf>
        </ItemGroupDef>
```

4.3 Dataset Variable Definitions

The CDISC SDTM, SEND, and ADaM specifications each define variable metadata requirements. Define-XML represents variable metadata using an ItemDef element. A detailed specification is provided in Section 5.3.12, ItemDef Element. Variables are associated with datasets through ItemRefs contained in ItemGroupDef elements. Additional variable metadata is associated with ItemDefs and ItemRefs using the elements listed in the following table.

Metadata	Define-XML Element	Reference
Controlled Terminology	CodeList	ItemDef/CodeListRef/@CodeListOID
Value-Level Metadata	def:ValueListDef	ItemDef/def:ValueListRef/@ValueListOID
Computational Method	MethodDef	ItemRef/@MethodOID
Comments	def:CommentDef	ItemDef/@def:CommentOID
Origin	def:Origin	ItemDef/def:Origin

Note: Variables can have both a CodeList and a ValueList attached because these provide semantically distinct information. A CodeList provides a list of allowable value. A ValueList is used to define metadata based on the value of another variable (Value-Level Metadata) to support data review and analysis in cases where variable metadata is not sufficient. See Section 4.5, Value-level Metadata Definitions, for further details.

4.3.1 Data Type Considerations

CDISC SDTM, SEND and ADaM variable data types are specified as either character or numeric ("Char" and "Num", respectively). Define-XML, because it is based on the CDISC ODM, supports a richer set of data types. The following table illustrates the mappings between Define-XML and SDTM data types that support the SDTMIG and SENDIG.

Define-XML Data Type	Submission Data Type	Length	Considerations
text	Char	Maximum allowable length	SAS v.5 Transport files restrict variable lengths to 200 characters.
integer	Num	The largest allowable integer width; note that for negative integers, the length will include the minus sign	Use for numeric or equivalent variables that have discrete whole values (non-fractional); can be positive, negative, or zero. ADaM numeric date variables are provided as integers.
float	Num	The largest allowable whole number width plus the maximum number of decimal digits	Use for numeric variables that may contain a fractional component. It represents the set of all the decimal numbers with arbitrary lengths.
datetime	Char	N/A	Use if values for SDTM or SEND variable represent Date Times (YYYY- MM-DDTHH:MM:SS).
date	Char	N/A	Use if values for SDTM or SEND variable represent complete (YYYY-MM-DD) dates.
time	Char	N/A	Use if values for SDTM or SEND variable represent complete (HH:MM:SS) times in ISO-8601 format.

partialDate	Char	N/A	Use when date values for SDTM or SEND may be right-truncated (i.e., the day or the day and month may be missing).
partialTime	Char	N/A	Use when time values for SDTM or SEND may be right-truncated (i.e., the seconds or seconds and minutes may be missing).
partialDatetime	Char	N/A	Use when date or time values for SDTM or SEND may be right-truncated; see partialDate and partialTime. For example, use to represent a full date and time in HH:MM.
incompleteDatetime	Char	N/A	Use when date values for SDTM or SEND may be missing a component but is not a partialDatetime or partialDate.
durationDatetime	Char	N/A	Use to indicate values for SDTM or SEND use the ISO8601 P notation to indicate a duration value.
intervalDatetime	Char	N/A	Use to indicate that the exact date is unknown but is known to be between 2 dates. Either of the dates may be expressed as a partialDate, partialDatetime, or durationDatetime.

Note: For SDTM and SEND, the date and time data types represent the planned specificity of the collected data. In the observed data, a variable defined with the Data Type partialDate may have a complete date but a variable defined with the Data Type Date may not be a partialDate.

4.3.2 Origin/Source/Traceability Considerations

The def:Origin element is used to provide traceability metadata for SENDIG, SDTMIG, and ADaM data.

The def:Origin element includes the Type and Source attributes. The Type attribute indicates how the data originated. The Source attribute identifies the party responsible for the data's origin.

The terminology used for the Type and Source attributes is expected to be managed and published with CDISC Controlled Terminology in non-extensible codelists. Users are encouraged to refer to the most recent version of CDISC Controlled Terminology.

In advance of CDISC publication of controlled terminology, the terms in the following tables can be used. Note that not all terms are applicable to all dataset types. These terms, and the terms published in CDISC Controlled Terminology, supersede terms which have been defined in other CDISC implementation guides.

Type	Definition
Collected	A value that is actually observed and recorded by a person or obtained by an instrument. Note that a collected entry translated to a synonymous controlled term still has a type Collected.
Derived	A value that is calculated by an algorithm or reproducible rule, and which is dependent upon other data values, including data values available within the dataset or externally provided data values. MethodDef must be used to document the algorithm or rule used for a derived value.
Assigned	Data that is either: Determined by individual judgment as provided by an evaluator, or Coded terms supplied as part of a coding process, or Values set independently of any subject-related data value in order to complete a dataset.
Protocol	Data that is defined as part of the study protocol, investigator instructions, standard operating procedures or trial design preparation
Predecessor	An entry that is copied from a variable in another dataset. The Description child element identifies the dataset and variable that is copied.

Not Available	Used when the origin is not available and cannot be determined. Sponsors should specify additional details that may be helpful to the reviewer in the Comments section of the data definition file.

4.3.2.1 Origin for SDTM Datasets

For SDTM datasets, both Type and Source attributes are required, except in the case of Predecessor where the Source attribute is not used.

All Type entries are valid except Not Available. Predecessor, however, is not frequently used. All Source entries are also valid.

The following table lists the allowed Origin Type and Source values for SDTM datasets. Type and Source values that do not apply to SDTM datasets are not listed. Table cells include examples that highlight cases where the specific combination of Type and Source attributes apply. Cells that include only an **X** indicate that the combination of Type and Source attributes is not valid. The Notes provide additional information about use of the Origin attributes for SDTM datasets.

Type	**Source**				**Notes**
	Subject	**Investigator**	**Vendor**	**Sponsor**	
Collected	ePro	CRF	Lab data, ECG	X	This term should be used for clinical data that were actually observed or recorded by a person or received from an instrument; it should not be used for data that have been interpreted, calculated, or derived from other information.
Derived	X	X	Lab data, ECG	SDTM	Derivation examples include calculations performed during data collection (e.g., --DY). Other derivation examples: calculations within ePRO (e.g., questionnaire section scores) and calculations within EDC (e.g., BMI, BSA).
Assigned	X	X	Adjudicator	SDTM	Examples of this include third-party attributions by an adjudicator, coded terms that are supplied as part of a coding process, and values that are set independently of any subject-related data values in order to complete SDTM fields such as DOMAIN and --TESTCD
Protocol	X	X	X	SDTM	An example would be VSPOS (Vital Signs Position), which could be specified in the protocol and be provided by other means (e.g. CRF, eDT).
Predecessor	X	X	X	X	Use when a value is an exact copy of another value in an SDTM dataset.

4.3.2.2 Origin for ADaM Datasets

For ADaM datasets, both Type and Source attributes are required, except in the case of Predecessor where the Source attribute is not used. For legacy analysis datasets, attributes Type and Source should be provided if the information is available.

The Type is usually Derived for data variables whose value is determined by a computational algorithm, Predecessor for variables that are a copy of another SDTM or ADaM variable, or Assigned for data variables with values assigned using a codelist. Collected and Protocol are generally not used, and Not Available is not valid.

The Source is always “Sponsor” for ADaM Derived and Assigned origin types. When Origin is Predecessor, the Source attribute is not used and the def:Origin has a child element Description that contains a reference to the dataset variable from which the value was copied, in the form <dataset name>.<variable name>; this is shown in Example 4.3.3.5.1, ADaM Variable with Predecessor Origin.

The following table lists the allowed Origin Type and Source values for ADaM datasets. Type and Source value combinations that do not apply to ADaM datasets are not listed. Cells that include only an **X** indicate that the combination of Type and Source attributes is not valid. The Notes provide additional information about use of the Origin attributes for ADaM datasets.

Type	Source	Notes
	Sponsor	
Derived	ADaM	For sponsor-performed analysis derivations in ADaM
Assigned	ADaM	
Predecessor	**X**	Use when a value is an exact copy of another value in either SDTM or ADaM dataset.

4.3.2.3 Origin for Nonclinical Datasets

For SEND datasets, only the Type attribute is used. All entries may be used, although use of Predecessor is not typical. Not Available is used only for Legacy datasets when the origin is unknown (not provided by the organization responsible for the data) and cannot be reliably determined, and is thus expected to be phased out.

The Source attribute is not used. The SEND Trial Summary dataset defines both the Test Facility and Test Site(s) responsible for the data represented in the SEND datasets. For data collected at a Test Site, the --NAM variable in a findings dataset may be used to identify the specific test site at which data originate.

4.3.3 Examples of Variable Definitions

4.3.3.1 SDTM Variable with Origin "Collected"

Example 4.3.3.1.1 illustrates the use of the def:Origin child element to provide a reference to the part of the study's annotated case report form (aCRF) used to collect data for the BRTHDTC variable for a SDTM Demographics dataset.

- The ItemRef child element with the ItemOID attribute value IT.DM.BRTHDTC in the ItemGroupDef element with the OID IG.DM references the ItemDef element with the matching OID. Because there is no Value Level metadata defined for the BRTHDTC variable, the Origin metadata must be provided at the variable level.
- The Type attribute of the def:Origin child element of the ItemDef that defines the BRTHDTC variable is set to Collected; the Source attribute is set to Investigator.
- The def:Origin includes a def:DocumentRef child element with the attribute leafID. The value of the leafID attribute must match the ID attribute of a def:leaf element in the same Define-XML document. The def:leaf element provides information about the name and location of the aCRF PDF file.
- The def:PDFPageRef child element of the def:DocumentRef provides details about the specific pages of the aCRF document where the BRTHDTC data is collected. The information in the def:PDFPageRef element can be used by an XSL stylesheet to construct a hyperlink to the specific entry page. In this case, the Type attribute of the def:PDFPageRef element has the value PhysicalRef to indicate that the value of the PageRefs attribute represents the physical page number within the aCRF PDF file.

See Section 4.3.2, Origin/Source/Traceability Considerations, for information about how the Origin types can be used.

Example 4.3.3.1.1 SDTM Variable with Origin "Collected"

```
<!-- ItemGroup Definition (DM) -->
<ItemGroupDef OID="IG.DM"
    Domain="DM"
    Name="DM"
    Repeating="No"
    …
    def:ArchiveLocationID="LF.DM">
    <Description>
        <TranslatedText xml:lang="en">Demographics</TranslatedText>
    </Description>
    …
    <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence=”1”/>
    …
    <ItemRef ItemOID="IT.DM.BRTHDTC" OrderNumber="8" Mandatory="No"/>
    …
</ItemGroupDef>

…

<!-- Item Definition: Variable Level (BRTHDTC) -->
<ItemDef OID="IT.DM.BRTHDTC" Name="BRTHDTC" DataType="date"
SASFieldName="BRTHDTC">
    <Description>
        <TranslatedText xml:lang="en">Date/Time of Birth</TranslatedText>
    </Description>
    <def:Origin Type="Collected" Source="Investigator">
    <def:DocumentRef leafID="LF.acrf">
    <def:PDFPageRef PageRefs="6" Type="PhysicalRef"/>
    </def:DocumentRef>
    </def:Origin>
</ItemDef>
```

4.3.3.2 SDTM Variables that Are Domain Keys

Example 4.3.3.2.1 illustrates how the ItemRef element is used to identify the variables for a dataset and how the definition details for each variable are represented.

- Each definition of a variable in a dataset must be provided using an ItemDef element. The ItemDef shown in this example has the Name attribute value STUDYID and Description/TranslatedText child element Study Identifier, indicating that this is the definition for the SDTM STUDYID variable. The OID for the ItemDef has the value IT.STUDYID, which matches ItemOID for the first ItemRef child element in the ItemGroupDef.
- The Mandatory attribute is set to Yes for each of the 3 ItemRef elements shown. This indicates that each of these variables must appear in the dataset and must have each row populated with a non-null data value.
- The def:Structure attribute provides a text description of the dataset structure.
- The KeySequence attribute indicates that the variable referenced by associated ItemRef is part of the dataset key. The value of the attribute identifies the order of the variable within the key. Each variable that is part of the Domain Key must have a KeySequence attribute. If a Supplemental Qualifier variable is part of the Domain key, the KeySequence attribute will be provided in the Value Level Metadata.

Note:

- The Role attribute is not included in the definition of the STUDYID variable because this attribute is optional and the Role is known for variables in SDTM standard domains.
- The Study Identifier variable may be defined by 1 ItemDef that is referenced in the ItemGroupDef for each dataset that includes it.
- The ItemRef@KeySequence attribute applies to domain keys for SDTM and SEND and dataset keys, for ADaM.

Example 4.3.3.2.1 SDTM Variables that Are Domain Keys

```
<!-- Dataset Definition (DM) -->
<ItemGroupDef OID="IG.DM"
    Domain="DM"
    Name="DM"
    Repeating="No"
    def:Structure="One record per subject"
    …
    def:ArchiveLocationID="LF.DM"
    def:StandardOID="STD.SDTMIG-3.2">
    <Description>
        <TranslatedText xml:lang="en">Demographics</TranslatedText>
    </Description>
    …
    <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
KeySequence="1"/>
    <ItemRef ItemOID="IT.DM.DOMAIN" OrderNumber="2" Mandatory="Yes"/>
    <ItemRef ItemOID="IT.USUBJID" OrderNumber="3" Mandatory="Yes" KeySequence="2”
MethodOID="MT.USUBJID"/>
    …
</ItemGroupDef>

…

<!-- Item Definition: Variable Level (STUDYID) -->
<ItemDef OID="IT.STUDYID" Name="STUDYID" DataType="text" Length="7"
SASFieldName="STUDYID">
    <Description>
        <TranslatedText xml:lang="en">Study Identifier</TranslatedText>
    </Description>
    <def:Origin Type="Protocol"/>
</ItemDef>
```

4.3.3.3 SDTM-derived Variables

Example 4.3.3.3.1 illustrates the use of the ODM MethodDef element to document the algorithm used to derive values for the SESTDTC and SEENDTC variables. For more information about representing computational algorithm metadata data, see Section 4.7, Computational Method Definitions.

Example 4.3.3.3.1 Derived Variable

```
<!-- Dataset Definition (SE) -->
<ItemGroupDef OID="IG.SE"
    Domain="SE" Name="SE"
    …
    <Description>
        <TranslatedText xml:lang="en">Subject Elements</TranslatedText>
    </Description>
    …
    <ItemRef ItemOID="IT.SE.SESTDTC" OrderNumber="7" Mandatory="Yes"
KeySequence="3" MethodOID="MT.SESTDTC"/>
    <ItemRef ItemOID="IT.SE.SEENDTC" OrderNumber="8" Mandatory="No"
KeySequence="4" MethodOID="MT.SEENDTC"/>
    …
</ItemGroupDef>
```

```
…
<!-- Item Definition: Variable Level (SEENDTC) -->
<ItemDef OID="IT.SE.SEENDTC" Name="SEENDTC" DataType="date"
SASFieldName="SEENDTC">
    <Description>
        <TranslatedText xml:lang="en">End Date/Time of Element</TranslatedText>
    </Description>
    <def:Origin Type="Derived" Source="Sponsor"/>
</ItemDef>
…
<!-- Item Definition: Variable Level (SESTDTC) -->
<ItemDef OID="IT.SE.SESTDTC" Name="SESTDTC" DataType="date"
SASFieldName="SESTDTC">
    <Description>
        <TranslatedText xml:lang="en">Start Date/Time of Element</TranslatedText>
    </Description>
    <def:Origin Type="Derived" Source="Sponsor"/>
</ItemDef>
…
<!-- Method Definitions -->
<MethodDef OID="MT.SESTDTC" Name="Algorithm to derive SESTDTC" Type="Computation"
>
    <Description>
        <TranslatedText xml:lang="en"> If Element = SCREEN, derived from SVSTDTC
where VISIT = SCREENING or from DS where DSDECOD = 'INFORMED CONSENT', whichever
is earliest. If Element = EOS, derived from DS where DSCAT = DISPOSITION EVENT.
For treatment Elements, derived from first EXSTDTC for the
element.</TranslatedText>
    </Description>
</MethodDef>
<MethodDef OID="MT.SEENDTC" Name="Algorithm to derive SEENDTC"
Type="Computation">
    <Description>
        <TranslatedText xml:lang="en"> If Element = SCREEN, derived from SVENDTC
where VISIT = SCREENING. If Element = EOS, derived from DS where DSCAT =
DISPOSITION EVENT or from the latest EXENDTC from EX whichever is later. For
treatment Elements, derived from last EXENDTC for the element. </TranslatedText>
    </Description>
</MethodDef>
```

4.3.3.4 ADaM Variable with Predecessor Origin

Example 4.3.3.4.1 shows how to represent the origin of an ADaM variable as a predecessor.

- The Name attribute of the ItemGroupDef is ADQSDADAS and the def:Class Name attribute indicate that this is an ADaM BDS dataset. The ItemRef ItemOID attribute matches the OID for the ItemDef for the Planned Treatment ADaM variable.
- The ItemDef element includes a def:Origin child element with the Type attribute value set to Predecessor. The Description child element of the def:Origin element has a TranslatedText child element with the value ADSL.TRT01P, indicating that the predecessor variable for the TRTP variable in the ADQSADAS dataset is the TRT01P variable in the ADSL dataset. Notice that the Source attribute of the def:Orgin element is not used in this case.
- When the date for an ADaM variable is taken from an SDTM supplemental qualifier dataset, the following notation should be used in the TranslatedText child element: <dataset name>.QVAL where QNAM="<QNAM value>". In general, a where condition should be specified when the values for the ADaM variable are taken from a subset of the referenced source dataset only.

Example 4.3.3.4.1 Linking an ADaM Variable to a Predecessor Using Origin

```
<ItemGroupDef OID="IG.ADQSADAS"
    Name="ADQSADAS"
    Repeating="Yes"
    IsReferenceData="No"
    Purpose="Analysis"
    def:Structure="One record per subject per parameter per analysis visit per
analysis date"
    def:CommentOID="COM.ADQSADAS"
    def:ArchiveLocationID="LF.ADQSADAS">
    <Description>
        <TranslatedText xml:lang="en">ADAS-Cog Analysis</TranslatedText>
    </Description>
    …
    <ItemRef ItemOID="IT.ADQSADAS.TRTP" OrderNumber="7" Mandatory="No"/>
    …
    <def:Class Name="BASIC DATA STRUCTURE"/>
<ItemGroupDef>

…

<!-- Item Definition: Variable Level (TRTP) -->
<ItemDef OID="IT.ADQSADAS.TRTP" Name="TRTP" DataType="text" Length="20">
    <Description>
        <TranslatedText xml:lang="en">Planned Treatment</TranslatedText>
    </Description>
    <CodeListRef CodeListOID="CL.ARM"/>
    <def:Origin Type="Predecessor">
      <Description>
        <TranslatedText xml:lang="en">ADSL.TRT01P</TranslatedText>
      </Description>
    </def:Origin>
</ItemDef>
```

4.3.3.5 ADaM-derived Variables

Example 4.3.3.5.1 shows how computational algorithms for ADaM-derived variables may be represented.

- The value of the Purpose attribute for the ItemGroupDef element is Analysis and the def:StandardOID attribute value matches the OID of the def:Standard for ADaMIG v1.1. The def:Class Name attribute indicates this is an ADaM BDS dataset.
- The ItemRef element shown here has the ItemOID attribute value that matches the OID for the ItemDef for the Change from Baseline variable. The MethodOID attribute on the ItemRef matches the OID for the MethodDef element at the bottom of the code sample.
- The ItemDef element includes a def:Origin child element with the Type attribute value "DERIVED". The def:Origin Source attribute has the value Sponsor because the study sponsor is responsible for the derivation programming.
- The Type attribute for the MethodDef element is Computation and the Description/TranslatedText child element provides a short description of how the change from baseline computation was performed when the dataset was created.

For more detailed information about representing computational algorithm metadata, see Section 4.7, Computational Method Definitions.

Example 4.3.3.5.1 ADaM-derived Variable

```
<def:Standards>
    <def:Standard OID="STD.ADaMIG-1.1" Name="ADaMIG-1.1" Type="IG" Version="1.1"
Status="Final"/>
    …
</def:Standards>
<!-- ItemGroup Definition (ADQSDAS) -->
<ItemGroupDef OID="IG.ADQSADAS"
    Name="ADQSADAS"
    Repeating="Yes"
    IsReferenceData="No"
    Purpose="Analysis"
    def:Structure="One record per subject per parameter per analysis visit per
analysis date"
    def:CommentOID="COM.ADQSADAS"
    def:ArchiveLocationID="LF.ADQSADAS"
    def:StandardOID="STD.ADaMIG-1.1">
    <Description>
        <TranslatedText xml:lang="en">ADAS-Cog Analysis</TranslatedText>
    </Description>
    …
    <ItemRef ItemOID="IT.ADQSADAS.CHG" OrderNumber="29" Mandatory="No"
MethodOID="MT.ADQSADAS.CHG"/>
    …
    <def:Class Name="BASIC DATA STRUCTURE"/>
    <def:leaf ID="LF.ADQSADAS" xlink:href="adqsadas.xpt">
        <def:title>adqsadas.xpt</def:title>
    </def:leaf>
</ItemGroupDef>
…
<!-- Item Definition: Variable Level (AVISIT) -->
<ItemDef OID="IT.ADQSADAS.CHG" Name="CHG" DataType="integer" Length="8">
    <Description>
        <TranslatedText xml:lang="en">Change from Baseline</TranslatedText>
    </Description>
    <def:Origin Type="Derived" Source="Sponsor"/>
</ItemDef>
…
<!-- Method Definition: Algorithm description -->
<MethodDef OID="MT.ADQSADAS.CHG" Name="CM.ADQSADAS.CHG" Type="Computation">
    <Description>
        <TranslatedText xml:lang="en">AVAL - BASE</TranslatedText>
    </Description>
</MethodDef>
```

4.3.3.6 Variable Using Alias for a Long SAS Name

This example shows a variable definition where the SAS name of the variable is longer than 8 characters. A use case for this is the situation where the Define-XML standard is used to transport dataset metadata of a legacy study in which the data format is in Dataset-XML.

Example 4.3.3.6.1 Variable with Long SAS Name

```
<ItemDef OID="IT.LB.SODIUM_NRIND" Name="SODIUM_NRIND" DataType="text" Length="7">
  <Description>
     <TranslatedText xml:lang="en">Sodium outside reference
range</TranslatedText>
```

```
    </Description>
    <CodeListRef CodeListOID="CL.ALTNRIND" />
    <Alias Context="SAS" Name="SODIUM_NRIND" />
    <def:Origin Type="Collected" Source="Sponsor">
       <def:DocumentRef leafID="LF.ACRF">
          <def:PDFPageRef PageRefs="19" Type="PhysicalRef"/>
       </def:DocumentRef>
    </def:Origin>
  </ItemDef>
```

4.4 Controlled Terminology Definitions

Controlled terminology, in the context of a study, refers to a set of standardized expressions or allowable values for data items in the study. A *codelist* is a unique subset of the controlled terminology assigned to 1 or more variables. A codelist can be defined at the variable or value level.

Beginning with SDTM v1.2, the SDTMIG and SENDIG require controlled terminology for many SDTM variables. For some variables, sponsor-specific controlled terminology may be recommended. All controlled terminology used in a study must be provided within the Define-XML document. Each codelist referenced by a study item shall be represented in the Define-XML document using a CodeList element.

- The CodeList element can define either an internal or an external codelist. Internal codelists include a list of allowable codes and, if applicable, their corresponding decodes. In most cases where controlled terminology is just an enumeration of allowed values, the EnumeratedItem element can be used to define the list of values. In cases where it is useful to provide decodes for the coded values, the CodeListItem element can be used.
- Codelists provided by third parties are specified using an ExternalCodeList element that identifies the dictionary by name and version. The href attribute contains the URL to the third-party dictionary, if available. **Note:** ItemDefs with date- and time-related datatypes cannot use codelists, and cannot reference the ISO 8601 standard using ExternalCodeLists. The ISO 8601 standard is not a controlled terminology; it is an international formatting standard for dates, times, datetimes, and durations.
- With few exceptions, coded values for CDISC codelists are provided as uppercase text. For variables whose definitions reference CodeList elements, the variable data content is expected to **exactly** match a single item in the corresponding CodeList, **including case**.

For codelists with entries that have a numeric significance, this can be described using the Rank attribute. For example, a list of enumerated text values including "Low", "Medium", and "High" might include a Rank attribute on each with values 1, 2, and 3, respectively. This can be used to override the normal lexical ordering of the entries with one based on numeric significance. The Rank attribute should not be used to define a display order.

The OrderNumber attribute can be used to define a display order for the items in a codelist.

CodeLists should never include a blank coded value. If an item does not require a value, the Mandatory attribute in the ItemRef element must be set to "No". Required items have Mandatory set to "Yes" and cannot have blank values.

CDISC codelists can be defined as *extensible*, which means that controlled terms may be added to the codelist. Sponsors may add to extensible codelists as long as they are not adding duplicates or synonyms of existing terms.

- If a CDISC codelist is not extensible, the expectation is that sponsors will use only the published list of terms.
- If a Define-XML CodeList element includes definitions of terms that are not included in the published list, then the def:ExtendedValue attribute should be set to "Yes" regardless of whether the codelist is extensible.

A codelist definition may include 1 or more Alias elements in order to facilitate the identification of the codelist components in an external system.

- For CodeList elements that define the contents of a CDISC codelist, an Alias element is used to provide the C-Codes that are used to identify codelists and coded terms within the National Cancer Institute's Enterprise Vocabulary System (https://evs.nci.nih.gov). By convention, the Alias Context attribute is set to "nci:ExtCodeId" in this scenario.
- For cases where the Define-XML includes multiple sponsor-defined subsets of the same parent CDISC Controlled Terminology CodeList, the Name attribute of the subset CodeList element should begin with the Name of parent CodeList. Each CodeList subset will contain the same C-Code in the CodeList Alias Name attribute.

Note: The controlled terminology examples are based on the CDISC Controlled Terminology available at the time of publication of this specification. Refer to CDISC Controlled Terminology publications (https://www.cancer.gov/research/resources/) for current codelists and terms.

4.4.1 Examples of Controlled Terminology Definitions

4.4.1.1 CodeList with EnumeratedItem Child Elements and Aliases

Example 4.4.1.1.1 illustrates the use of a CDISC codelist. Aliases are included at both the CodeList and EnumeratedItem levels.

- This CodeList represents a CDISC Controlled Terminology, so the CodeList element includes a def:StandardOID attribute that matches the OID for a def:Standard that identifies a CDISC Controlled Terminology version and publishing set. The def:Standard is not shown here but can be seen in Example 4.1.2.1.1, Single SDTMIG and Controlled Terminology Reference.
- At the CodeList level, the Alias Context attribute references "nci:ExtCodeID" and the Name attribute contains the C-Code for the codelist.
- At the EnumeratedItem level, the Alias Context attribute references “nci:ExtCodeID” and the Name attribute contains the C-Code for the term.

Note: The location of the Alias element at the CodeList level follows all CodeList EnumeratedItem child elements.

Example 4.4.1.1.1 CodeList with Alias Elements for C-Codes

```
<CodeList OID="CL.ACN" Name="Action Taken with Study Treatment" DataType="text"
 def:StandardOID="STD.CT-20141219">
    <EnumeratedItem CodedValue="DOSE NOT CHANGED">
        <Alias Name="C49504" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="DOSE REDUCED">
        <Alias Name="C49505" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="DRUG INTERRUPTED">
        <Alias Name="C49501" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="DRUG WITHDRAWN">
        <Alias Name="C49502" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <Alias Name="C66767" Context="nci:ExtCodeID"/>
</CodeList>
```

4.4.1.2 CodeLists for VSTEST and VSTESTCD

Example 4.4.1.2.1 shows CodeLists for VSTEST and VSTESTCD variables. The VSTEST CodeList uses EnumeratedItems to store the list of permissible values, as published in the NCI/CDISC SDTM and SEND Controlled Terminology packages. The VSTESTCD CodeList uses CodeListItems to reflect the association with VSTEST items.

Example 4.4.1.2.1 CodeList Elements for VSTEST and VSTESTCD

```
<CodeList OID="CL.VSTEST" Name="Vital Signs Test Name" DataType="text"
def:StandardOID="STD.CT-20141219">
    <EnumeratedItem CodedValue="Body Frame Size">
        <Alias Name="C49680" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="Diastolic Blood Pressure">
        <Alias Name="C25299" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="Height">
        <Alias Name="C25347" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="Pulse Rate">
        <Alias Name="C49676" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="Systolic Blood Pressure">
        <Alias Name="C25298" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="Weight">
        <Alias Name="C25208" Context="nci:ExtCodeID"/>
    </EnumeratedItem>
    <Alias Name="C67153" Context="nci:ExtCodeID"/>
</CodeList>

<CodeList OID="CL.VSTESTCD" Name="Vital Signs Test Code" DataType="text"
SASFormatName="$VSTESTC" def:StandardOID="STD.CT-20141219">
    <CodeListItem CodedValue="DIABP">
        <Decode>
            <TranslatedText xml:lang="en">Diastolic Blood
Pressure</TranslatedText>
        </Decode>
        <Alias Name="C25299" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="FRMSIZE">
        <Decode>
            <TranslatedText xml:lang="en">Body Frame Size</TranslatedText>
        </Decode>
        <Alias Name="C49680" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="HEIGHT">
        <Decode>
            <TranslatedText xml:lang="en">Height</TranslatedText>
        </Decode>
        <Alias Name="C25347" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="PULSE">
        <Decode>
            <TranslatedText xml:lang="en">Pulse Rate</TranslatedText>
        </Decode>
        <Alias Name="C49676" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="SYSBP">
        <Decode>
            <TranslatedText xml:lang="en">Systolic Blood
Pressure</TranslatedText>
        </Decode>
        <Alias Name="C25298" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="WEIGHT">
```

```
        <Decode>
            <TranslatedText xml:lang="en">Weight</TranslatedText>
        </Decode>
        <Alias Name="C25208" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <Alias Name="C66741" Context="nci:ExtCodeID"/>
</CodeList>
```

4.4.1.3 CodeList with CodeListItem Child Elements

This example shows the definition of sponsor-specific controlled terminology for the SDTM variable ARMCD. Because it is sponsor-defined, the CodeList element includes a def:IsNonStandard attribute with the value "Yes". This terminology requires decoded values to facilitate data interpretation, so the CodeList element includes CodeListItem elements with a Decode element. Each Decode element, in turn, includes a TranslatedText element that contains the decoded value. The xml:lang attribute on the TranslatedText element is used to identify the language for the contained text. If the attribute is omitted, the local language defined in the browser displaying the Define-XML is assumed.

Note: The OrderNumber attribute allows displaying the elements in a different sequence than the order listed in the Define-XML document.

Example 4.4.1.3.1 CodeList with CodeListItem Elements

```
<CodeList OID="CL.ARMCD" Name="Planned Arm Code" DataType="text"
SASFormatName="$ARMCD" def:IsNonStandard="Yes">
  <CodeListItem CodedValue="PLACEBO" OrderNumber="3">
    <Decode><TranslatedText xml:lang="en">Placebo</TranslatedText></Decode>
  </CodeListItem>
  <CodeListItem CodedValue="SCRNFAIL" OrderNumber="4">
    <Decode><TranslatedText xml:lang="en">Screen
Failure</TranslatedText></Decode>
  </CodeListItem>
  <CodeListItem CodedValue="WONDER10" OrderNumber="1">
    <Decode><TranslatedText xml:lang="en">Miracle Drug 10
mg</TranslatedText></Decode>
  </CodeListItem>
  <CodeListItem CodedValue="WONDER20" OrderNumber="2">
    <Decode><TranslatedText xml:lang="en">Miracle Drug 20
mg</TranslatedText></Decode>
  </CodeListItem>
</CodeList>
```

4.4.1.4 CodeList Using Rank Attribute

In this example, controlled terminology for the SDTM variable AESEV is defined using the SDTM Codelist AESEV. The Rank attribute is provided for each CodeListItem to indicate the numeric significance .of the code relative to others in the CodeList.

Example 4.4.1.4.1 CodeList Using Rank Attribute

```
<CodeList OID="CL.AESEV" Name="Severity/Intensity Scale for Adverse Events"
DataType="text"
                                          SASFormatName="$AESEV"
def:StandardOID="STD.CT-20141219">
    <CodeListItem CodedValue="MILD" Rank="1">
        <Decode>
            <TranslatedText xml:lang="en">Grade 1</TranslatedText>
        </Decode>
        <Alias Name="C41338" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="MODERATE" Rank="2">
        <Decode>
            <TranslatedText xml:lang="en">Grade 2</TranslatedText>
        </Decode>
        <Alias Name="C41339" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="SEVERE" Rank="3">
        <Decode>
            <TranslatedText xml:lang="en">Grade 3</TranslatedText>
        </Decode>
        <Alias Name="C41340" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <Alias Name="C66769" Context="nci:ExtCodeID"/>
</CodeList>
```

4.4.1.5 CodeList with ExternalCodeList Child Element

In this example, controlled terminology for the SDTM variable AEDECOD is defined using a third-party dictionary (i.e., MedDRA). The contents for this terminology must be licensed through the Medical Dictionary for Regulatory Activities Maintenance and Support Services Organization; to facilitate interpretation of the clinical data, a CodeList element is defined using an ExternalCodeList element to specify the dictionary name and version.

Example 4.4.1.5.1 CodeList for External Dictionary Reference

```
<ItemDef OID="IT.AE.AEDECOD" Name="AEDECOD" DataType="text" Length="18"
SASFieldName="AEDECOD">
    <Description>
        <TranslatedText xml:lang="en">Dictionary-Derived Term</TranslatedText>
    </Description>
    <CodeListRef CodeListOID="CL.AEDICT_F"/>
    <def:Origin Type="Assigned"/>
</ItemDef>

<CodeList OID="CL.AEDICT_F" Name="Adverse Event Dictionary" DataType="text">
    <ExternalCodeList Dictionary="MedDRA" Version="14.0"/>
</CodeList>
```

4.4.1.6 CodeList for CDISC Controlled Terminology with an Extended CodeListItem

In Example 4.4.1.6.1, the CodeListItem SUBJINIT has been added by the sponsor to extend external controlled terminology. The last CodeListItem element does not include an Alias element because it represents a value added to an extensible CDISC Controlled Terminology. The final Alias element identifies the CDISC Controlled Terminology for "Subject Characteristic Test Code (SCTESTCD)".

Example 4.4.1.6.1 CDISC Controlled Terminology with an Extended Item

```
<CodeList OID="CL.SCTESTCD" Name="Subject Characteristic Code (SCTESTCD)"
DataType="text"
        SASFormatName="$SCTESTC"  def:StandardOID="STD.CT-20141219">
    <CodeListItem CodedValue="EDLEVEL">
        <Decode>
            <TranslatedText xml:lang="en">Education Level</TranslatedText>
        </Decode>
        <Alias Name="C17953" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="MARISTAT">
        <Decode>
            <TranslatedText xml:lang="en">Marital Status</TranslatedText>
        </Decode>
        <Alias Name="C25188" Context="nci:ExtCodeID"/>
    </CodeListItem>
    <CodeListItem CodedValue="SUBJINIT" def:ExtendedValue="Yes">
        <Decode>
            <TranslatedText xml:lang="en">Subject Initials</TranslatedText>
        </Decode>
    </CodeListItem>
    <Alias Name="C74559" Context="nci:ExtCodeID"/>
</CodeList>
```

4.4.1.7 Multiple Subsets of CDISC Controlled Terminology

In this example, 2 CodeList elements (Name=”Frequency, Study Subset to be used in EX.EXDOSFRQ” and Name=”Frequency, Study Subset to be used in SU.SUDOSFRQ”) have been added by the sponsor to differentiate the allowed frequency terms for the --DOSFRQ variable in the SDTMIG Exposure (EX) and Substance Use (SU) domains, respectively.

- The Decode values shown here are taken from the CDISC Synonym field in the CDISC Controlled Terminology publication. The implementation guides for some CDISC standards may recommend use of specific Decode values. In other cases, the choice of the decode is at the sponsor's discretion.
- The child Alias element for each CodeList element refers to the C-Code “C71113” for the Name=”Frequency” CodeList. The version of the Controlled Terminology is identified in the def:Standard element where the OID attribute value is STD.CT-20141219.

Example 4.4.1.7.1 CDISC Controlled Terminology References, Multiple Subsets

```
<CodeList OID="CL.FREQ_EX" Name="Frequency, Study Subset to be used in
EX.EXDOSFRQ" DataType="text"
                                               SASFormatName="$FMT0049"
def:StandardOID="STD.CT-20141219">
  <CodeListItem CodedValue="Q24H" OrderNumber="1">
   <Decode>
      <TranslatedText xml:lang="en">Every 24 hours</TranslatedText>
     </Decode>
    <Alias Context="nci:ExtCodeID" Name="C64515"/>
  </CodeListItem>
  <Alias Context="nci:ExtCodeID" Name="C71113"/>
</CodeList>
<CodeList OID="CL.FREQ_SU" Name="Frequency, Study Subset to be used in
SU.SUDOSFRQ" DataType="text"
```

```
                                                   SASFormatName="$FMT0050"
def:StandardOID="STD.CT-20141219">
  <CodeListItem CodedValue="EVERY WEEK" OrderNumber="1">
     <Decode>
      <TranslatedText xml:lang="en">Every week</TranslatedText>
     </Decode>
     <Alias Context="nci:ExtCodeID" Name="C67069"/>
   </CodeListItem>
   <CodeListItem CodedValue="PA" OrderNumber="2">
     <Decode>
       <TranslatedText xml:lang="en">Per Year</TranslatedText>
    </Decode>
     <Alias Context="nci:ExtCodeID" Name="C74924"/>
   </CodeListItem>
   <CodeListItem CodedValue="QD" OrderNumber="3">
    <Decode>
      <TranslatedText xml:lang="en">Daily</TranslatedText>
     </Decode>
     <Alias Context="nci:ExtCodeID" Name="C25473"/>
   </CodeListItem>
   <CodeListItem CodedValue="QM" OrderNumber="4">
    <Decode>
      <TranslatedText xml:lang="en">Every Month</TranslatedText>
     <Alias Context="nci:ExtCodeID" Name="C64498"/>
   </CodeListItem>
   <Alias Context="nci:ExtCodeID" Name="C71113"/>
 </CodeList>
```

4.5 Value-level Metadata Definitions

The normalized data structure used by datasets based on the SDTM, SEND, and ADaM (generally 1 record per subject per test code per visit or observation) provides an efficient method for transmitting information. However, there are cases where the dataset variable metadata does not provide sufficient detail to support data review and analysis. In these cases, value-level metadata should be provided in the Define-XML document. Value-level metadata enables the specification of the metadata of a variable under conditions involving 1 or more dataset variables. The definition of a variable for a specific condition is known as value-level metadata.

Value-level metadata is specified in Define-XML using def:ValueListDef elements (see Section 5.3.9, def:ValueListDef Element, for a detailed specification).

- Value list definitions (or valuelists) are linked with variable definitions by including a def:ValueListRef element in the ItemDef element that defines the variable.
- Note that a dataset can include multiple variables with valuelist definitions. (See Section 5.3.12.2, def:ValueListRef Element, for a detailed specification.)

Value-level metadata should be provided when there is a need to describe differing metadata attributes for subsets of cells within a column.

- It is most often used within SDTM Findings domains to provide definitions for Variables (e.g., --ORRES, --ORRESU, --STRES, --STRESU) that are specific to each test code (value of --TESTCD). It is not required for Findings domains where the results have the same characteristics in all records (e.g., Inclusion Exclusion domains).
- In ADaM, value-level metadata often describes AVAL or AVALC in BDS data based on values of PARAMCD, but can be applied in other cases where useful.

Value-level metadata can also be used on all classes of SDTM domains. For example, in a Dispositions domain, value-level metadata might define the codelists for DSTERM and DSDECOD for each value of DSCAT.

Value-level metadata for a SDTM SUPPQUAL dataset can be used to include one or more Supplemental Qualifier variables in the Domain key.

Value-level metadata should be applied when it provides information useful for interpreting study data. It need not be applied in all cases.

- As an example, the --TEST variable could either (1) be specified by a single variable with a codelist containing all the test names, or (2) have value-level metadata specifying exactly which test name is appropriate for each --TESTCD. Although both of these approaches are valid, the value-level metadata approach is more complicated and may not provide any information that will benefit a consumer of the data. It is left to the discretion of the creator when it is useful to provide value-level metadata and when it is not. The creator should seek guidance from the consumer.
- When deciding whether to define value-level metadata, a good rule of thumb is that if providing value-level metadata for a variable would mean each value has a codelist containing only 1 CodeList Item, it is probably more appropriate to define a codelist rather than a valuelist. This is the case with --TEST and --TESTCD variables.
- Similarly, if the values of the DataType and Length attributes of each value are the same and there is no codelist involved, it is not necessary to provide value-level metadata.

Variable-level metadata applies to all cells in a column within a table unless superseded by **value**-level metadata.

- For value-level definitions that do not have a specified property provided, the one given at the variable level may apply. Note that (1) some properties are required to be specified for each item definition and cannot be left blank; (2) other properties cannot be redefined at value level unless the value-level definition corresponds to a non-standard variable definition (e.g. KeySequence, Role).
- Value-level definitions must be compatible with the parent variable definition. For example, no value-level length can exceed the length of the parent variable.
- For some other properties, the property can be redefined without compatibility issues at all; for example, in case there is a method provided at the variable level, value-level definitions could refer to a different method than the one specified at the variable level.
- If a property not required to be specified for each item definition (e.g., a codelist reference) is specified in the variable-level metadata, there is no need to provide it at the value level as well. If codelists are provided at both levels, the codelist defined at the Variable level is expected to be a superset of all codelists specified in the value-level definitions for the specified variable.

Example 4.5.1 Variable Definition Referencing a ValueList Definition

```xml
<ItemDef OID="IT.VS.VSORRES" Name="VSORRES" DataType="text" Length="30"
SASFieldName="VSORRES">
    <Description>
        <TranslatedText xml:lang="en">Result or Finding in Original
Units</TranslatedText>
    </Description>
    <def:Origin Type="CRF">
        <def:DocumentRef leafID="LF.blankcrf">
            <def:PDFPageRef PageRefs="11" Type="PhysicalRef"/>
        </def:DocumentRef>
    </def:Origin>
    <def:ValueListRef ValueListOID="VL.VS.VSORRES"/>
</ItemDef>
```

Example 4.5.2 may be used to define the DIABP original result's Value within the VSORRES column. In this example, VSORRES has a length of 30 and can hold any text value, whereas VSORRES values for DIABP records must be integers of length no greater than 2.

Example 4.5.2 Value Definition

```
<ItemDef OID="IT.VS.VSORRES.DIABP" Name="DIABP" DataType="integer" Length="2"
SASFieldName="DIABP">
    <Description>
        <TranslatedText xml:lang="en">Diastolic Blood Pressure</TranslatedText>
    </Description>
</ItemDef>
```

4.5.1 Where Clauses for Value-level Metadata

Where Clauses are used to describe the conditions under which the definition of a value applies in a machine-readable form. Each Value definition may have a Where Clause attached to it to describe when that Value applies. The Where Clause mechanism allows the definition of slices. *Slices* are subsets of a dataset that typically include a subset of the dataset rows which share similar metadata.

Where Clauses can only contain references to:

1. Variables in the current dataset
2. Variables in other subject-level datasets within the same metadata version in a Define-XML document. In this case, the implied dataset join should be documented in a comment attached to the WhereClauseDef.

Note: In the current specification, there is no mechanism for machine-readable join specifications. Therefore, the Comment functionality is intended to document the implied join.

Where Clauses define a condition by using one or more Range Checks.

- Where there are multiple Range Checks, the condition is defined as the logical AND of all the Range Checks.
- If there are multiple Where Clauses, the condition is defined as the logical OR of the Where Clauses.

Example 4.5.1.1 illustrates construction of compound Where Clauses.

Example 4.5.1.1 Where Clauses

```
<!--  Where Clause definitions for:
      Where VSTESTCD = 'SYSBP' and VSPOS = 'SITTING' -->
<def:WhereClauseDef OID="WC.VS.VSTESTCD.SYSBP.VS.VSPOS.SITTING">
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
        <CheckValue>SYSBP</CheckValue>
    </RangeCheck>
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSPOS" Comparator="EQ">
        <CheckValue>SITTING</CheckValue>
    </RangeCheck>
</def:WhereClauseDef>
<!--  Where Clause definitions for:
      Where VSTESTCD = 'WEIGHT' and COUNTRY WITH METRIC SYSTEM -->
<def:WhereClauseDef OID="WC.VS.VSTESTCD.WEIGHT.[DM].COUNTRY.CMETRIC"
def:CommentOID="COM.SUBJECTDATA-JOIN-DM">
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
        <CheckValue>WEIGHT</CheckValue>
    </RangeCheck>
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.DM.COUNTRY" Comparator="IN">
        <CheckValue>CAN</CheckValue>
        <CheckValue>MEX</CheckValue>
    </RangeCheck>
</def:WhereClauseDef>
```

4.5.2 Examples of Value-level Metadata Definitions

It is important to understand that only one XML representation of value-level metadata exists. However, to facilitate understanding of these concepts, this document discusses value-level metadata in terms of both valuelists and slices. Valuelists and slices are simply different ways of visualizing the same underlying metadata: Whereas valuelists show the definitions for a single variable for each condition, slices show the definitions for a whole domain for a given condition (i.e., Where Clause). More details about the distinction between these visualizations are illustrated in the examples in Appendix B, Visualizing Value-Level Metadata.

4.5.2.1 Value-level Metadata: Vital Signs Domain

Because the VS domain is a Findings domain, the attributes of variables VSORRES, VSORRESU, VSSTRES, VSSTRESU (and other variables) can have different values depending on the value of the VSTESTCD and values of other variables. Therefore, a description of all the different possible values for those variables at a more granular level is appropriate.

Example 4.5.2.1.1 illustrates how to provide value-level metadata for the VS domain. In this example, only variables VSORRES and VSORRESU are shown with a valuelist attached to them. In contrast, variable VSSTRESU is illustrated only with a codelist that covers all the possible values.

This example includes:

- Two def:ValueListDef elements. The first provides 6 value definitions for the VSORRES variable, the second provides 4 value definitions for the VSORRESU Variable.
- Two ItemDef elements for the VSORRES and VSORRESU variables. Each of these ItemDef elements includes a def:ValueListRef element.
- The ItemDef element that provides the value level definition for VSORRES for the VS Height observation.
- The ItemDef element that provides the value level definition for VSORRESU when the Height observation is collected in a metric country.
- The ItemDef element that provides the value level definition for VSORRESU when the Height observation is collected in a non-metric country.

In the XML each Value Definition includes an ItemRef element and a def:WhereClauseRef element. The ItemRef element identifies an ItemDef element and the def:WhereClauseRef element identifies a def:WhereClauseDef element.

Example 4.5.2.1.1 Value-level Metadata for SDTM Vital Signs

```
<!--  Value Level Metadata definitions -->
   <def:ValueListDef OID="VL.VS.VSORRES">
        <ItemRef ItemOID="IT.VS.VSORRES.DIABP" OrderNumber="1" Mandatory="Yes">
          <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.DIABP"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRES.FRMSIZE" OrderNumber="2" Mandatory="No">
          <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.FRMSIZE"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRES.HEIGHT" OrderNumber="3" Mandatory="Yes">
          <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.HEIGHT"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRES.PULSE" OrderNumber="4" Mandatory="No">
          <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.PULSE"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRES.SYSBP" OrderNumber="5" Mandatory="Yes">
          <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.SYSBP"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRES.WEIGHT" OrderNumber="6" Mandatory="Yes">
          <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.WEIGHT"/>
        </ItemRef>
      </def:ValueListDef>
      <def:ValueListDef OID="VL.VS.VSORRESU">
        <ItemRef ItemOID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CMETRIC"
OrderNumber="1" Mandatory="Yes">
          <def:WhereClauseRef
WhereClauseOID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CMETRIC"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CNMETRIC"
OrderNumber="2" Mandatory="Yes">
          <def:WhereClauseRef
WhereClauseOID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CNMETRIC"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRESU.WEIGHT.DM.COUNTRY.CMETRIC"
OrderNumber="3" Mandatory="Yes">
          <def:WhereClauseRef
WhereClauseOID="WC.VS.VSTESTCD.WEIGHT.[DM].COUNTRY.CMETRIC"/>
        </ItemRef>
        <ItemRef ItemOID="IT.VS.VSORRESU.WEIGHT.DM.COUNTRY.CNMETRIC"
OrderNumber="4" Mandatory="Yes">
          <def:WhereClauseRef
WhereClauseOID="WC.VS.VSTESTCD.WEIGHT.[DM].COUNTRY.CNMETRIC"/>
        </ItemRef>
      </def:ValueListDef>

<!--  Item definition for VSORRES -->
<ItemDef OID="IT.VS.VSORRES" Name="VSORRES" DataType="text" Length="200"
SASFieldName="VSORRES">
    <Description>
        <TranslatedText xml:lang="en">Result or Finding in Original
Units</TranslatedText>
    </Description>
    <def:ValueListRef ValueListOID="VL.VS.VSORRES"/>
</ItemDef>
<!--  Item definition for VSORRESU -->
<ItemDef OID="IT.VS.VSORRESU" Name="VSORRESU" DataType="text" Length="9"
SASFieldName="VSORRESU">
    <Description>
        <TranslatedText xml:lang="en">Standard Units</TranslatedText>
    </Description>
    <def:ValueListRef ValueListOID="VL.VS.VSORRESU"/>
</ItemDef>
...
<ItemDef OID="IT.VS.VSORRES.DIABP" Name="DIABP" DataType="integer" Length="2"
SASFieldName="DIABP">
    <Description>
        <TranslatedText xml:lang="en">Diastolic Blood Pressure</TranslatedText>
    </Description>
</ItemDef>
...
<ItemDef OID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CMETRIC" Name="HEIGHT"
DataType="float"
Length="5" SignificantDigits="2" SASFieldName="HEIGHT">
    <Description>
        <TranslatedText xml:lang="en">Height</TranslatedText>
    </Description>
    <CodeListRef CodeListOID="CL.UH_MC"/>
</ItemDef>
<ItemDef OID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CNMETRIC" Name="HEIGHT"
DataType="float"
Length="5" SignificantDigits ="2" SASFieldName="HEIGHT">
```

```
    <Description>
        <TranslatedText xml:lang="en">Height</TranslatedText>
    </Description>
    <CodeListRef CodeListOID="CL.UH_NMC"/>
</ItemDef>

<!--  Item definition for VSSTRESU -->
<ItemDef OID="IT.VS.VSSTRESU" Name="VSSTRESU" DataType="text" Length="9"
SASFieldName="VSSTRESU">
    <Description>
        <TranslatedText xml:lang="en">Standard Units</TranslatedText>
    </Description>
    <CodeListRef CodeListOID="CL.VSRESU"/>
    <def:Origin Type="Derived"/>
</ItemDef>
```

4.5.2.2 Where Clause Metadata: Vital Signs

Example 4.5.2.2.1 shows 3 Where Clauses, defining 3 slices or conditions for the VS domain:

- Where VSTESTCD = "HEIGHT" and COUNTRY IN ["CAN", "MEX"]
- Where VSTESTCD = "HEIGHT" and COUNTRY IN ["USA"]
- Where VSTESTCD = "DIABP"

As discussed in Section 4.5.1, Where Clauses for Value-level Metadata, each value definition requires its own Where Clause.

For VSORRES, a single DIABP Value definition is used regardless of the COUNTRY, so that the Value definition Where Clause has only a single condition, VSTESTCD="DIABP".

For VSORRESU, there are 2 possible definitions for HEIGHT depending on the value of COUNTRY. For this reason, a Value definition and corresponding Where Clause is provided for each condition.

Example 4.5.2.2.1 WhereClause for Vital Signs

```
<def:WhereClauseDef OID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CMETRIC"
def:CommentOID="COM.SUBJECTDATA-JOIN-DM">
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
        <CheckValue>HEIGHT</CheckValue>
    </RangeCheck>
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.DM.COUNTRY" Comparator="IN">
        <CheckValue>CAN</CheckValue>
        <CheckValue>MEX</CheckValue>
    </RangeCheck>
</def:WhereClauseDef>
<def:WhereClauseDef OID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CNMETRIC"
def:CommentOID="COM.SUBJECTDATA-JOIN-DM">
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
        <CheckValue>HEIGHT</CheckValue>
    </RangeCheck>
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.DM.COUNTRY" Comparator="EQ">
        <CheckValue>USA</CheckValue>
    </RangeCheck>
</def:WhereClauseDef>

<!--  Where Clause definitions for:  Where VSTESTCD = 'DIABP' -->
```

```
<def:WhereClauseDef OID="WC.VS.VSTESTCD.DIABP ">
    <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
        <CheckValue>DIABP</CheckValue>
    </RangeCheck>
</def:WhereClauseDef>

<!--  Documentation to join a subject-level dataset with the Demographics dataset
-->
<def:CommentDef OID="COM.SUBJECTDATA-JOIN-DM">
    <Description>
        <TranslatedText xml:lang="en">Join any Subject Level dataset with the
Demographics dataset based on [IG.datasetname]IT.USUBJID = [IG.DM]IT.USUBJID,
assuming 'IG.datasetname' is the OID of the ItemGroupDef that defines the
subject-level dataset to be joined with the Demographics
dataset.</TranslatedText>
    </Description>
</def:CommentDef>
```

4.5.2.3 Value-level Metadata for a SUPPQUAL Domain

This example illustrates Value-level definitions for variable QVAL in the SUPPLB and SUPPQS datasets as supplemental or non-standard variables for the Laboratory Test Results and Questionnaires domains.

Note: The definition of the valuelist is for the variable QVAL and not for the variable QNAM.

Example 4.5.2.3.1 Value-level Metadata: SUPPQUAL

```
<def:ValueListDef OID="VL.SUPPLB.QVAL">
    <ItemRef ItemOID="IT.SUPPLB.QVAL.LBCLSIG" OrderNumber="1" Mandatory="No"
MethodOID="MT.CLSIG" Role="Record Qualifier">
        <def:WhereClauseRef WhereClauseOID="WC.SUPPLB.QNAM.LBCLSIG"/>
    </ItemRef>
</def:ValueListDef>
<def:ValueListDef OID="VL.SUPPQS.QVAL">
    <ItemRef ItemOID="IT.SUPPQS.QVAL.RTRINIT" OrderNumber="1" Mandatory="No"
Role="Identifier">
        <def:WhereClauseRef WhereClauseOID="WC.SUPPQS.QNAM.RTRINIT"/>
    </ItemRef>
</def:ValueListDef>

<ItemDef OID="IT.SUPPLB.QVAL.LBCLSIG" Name="LBCLSIG" DataType="text" Length="1"
SASFieldName="LBCLSIG" Role="Record Qualifier">
    <Description>
        <TranslatedText xml:lang="en">Clinically Significant</TranslatedText>
    </Description>
    <CodeListRef CodeListOID="CL.NY"/>
    <def:Origin Type="Derived"/>
</ItemDef>

<ItemDef OID="IT.SUPPQS.QVAL.RTRINIT" Name="RTRINIT" DataType="text" Length="3"
SASFieldName="RTRINIT" Role="Result Qualifier"
def:CommentOID="COM.SUPPQS.QVAL.RTRINIT">
    <Description>
        <TranslatedText xml:lang="en">Rater Initials</TranslatedText>
    </Description>
    <def:Origin Type="CRF">
        <def:DocumentRef leafID="LF.blankcrf">
```

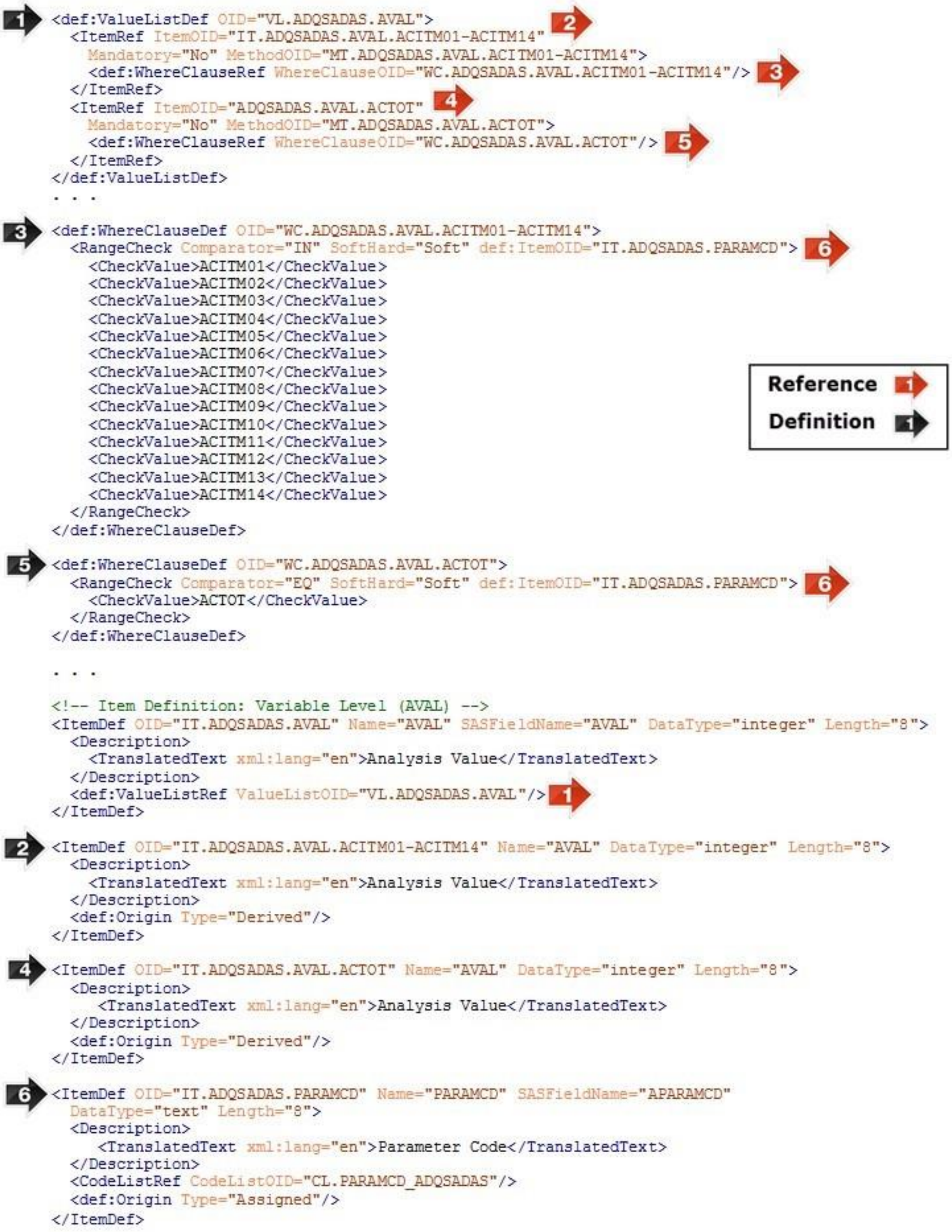
```
            <def:PDFPageRef PageRefs="13 14 17" Type="PhysicalRef"/>
        </def:DocumentRef>
    </def:Origin>
</ItemDef>
```

4.5.2.4 ADaM Parameter-level Metadata for a BDS Dataset

Example 4.5.2.4.1 illustrates the use of the def:ValueListDef with a reference to a def:WhereClauseRef used to define metadata for AVAL based on the value of the PARAMCD variable in an ADaM BDS dataset. This example illustrates the concepts of linking between a definition of an object (black arrows) and references to it (red arrows).

Note: Only 2 parameter-level definitions are required in this case because the metadata is the same for all parameter values except one.

Example 4.5.2.4 ADaM Parameter-level Metadata



```
<def:ValueListDef OID="VL.ADQSADAS.AVAL">
  <ItemRef ItemOID="IT.ADQSADAS.AVAL.ACITM01-ACITM14"
    Mandatory="No" MethodOID="MT.ADQSADAS.AVAL.ACITM01-ACITM14">
    <def:WhereClauseRef WhereClauseOID="WC.ADQSADAS.AVAL.ACITM01-ACITM14"/>
  </ItemRef>
  <ItemRef ItemOID="ADQSADAS.AVAL.ACTOT"
    Mandatory="No" MethodOID="MT.ADQSADAS.AVAL.ACTOT">
    <def:WhereClauseRef WhereClauseOID="WC.ADQSADAS.AVAL.ACTOT"/>
  </ItemRef>
</def:ValueListDef>
. . .

<def:WhereClauseDef OID="WC.ADQSADAS.AVAL.ACITM01-ACITM14">
  <RangeCheck Comparator="IN" SoftHard="Soft" def:ItemOID="IT.ADQSADAS.PARAMCD">
    <CheckValue>ACITM01</CheckValue>
    <CheckValue>ACITM02</CheckValue>
    <CheckValue>ACITM03</CheckValue>
    <CheckValue>ACITM04</CheckValue>
    <CheckValue>ACITM05</CheckValue>
    <CheckValue>ACITM06</CheckValue>
    <CheckValue>ACITM07</CheckValue>
    <CheckValue>ACITM08</CheckValue>
    <CheckValue>ACITM09</CheckValue>
    <CheckValue>ACITM10</CheckValue>
    <CheckValue>ACITM11</CheckValue>
    <CheckValue>ACITM12</CheckValue>
    <CheckValue>ACITM13</CheckValue>
    <CheckValue>ACITM14</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>

<def:WhereClauseDef OID="WC.ADQSADAS.AVAL.ACTOT">
  <RangeCheck Comparator="EQ" SoftHard="Soft" def:ItemOID="IT.ADQSADAS.PARAMCD">
    <CheckValue>ACTOT</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>

. . .

<!-- Item Definition: Variable Level (AVAL) -->
<ItemDef OID="IT.ADQSADAS.AVAL" Name="AVAL" SASFieldName="AVAL" DataType="integer" Length="8">
  <Description>
    <TranslatedText xml:lang="en">Analysis Value</TranslatedText>
  </Description>
  <def:ValueListRef ValueListOID="VL.ADQSADAS.AVAL"/>
</ItemDef>

<ItemDef OID="IT.ADQSADAS.AVAL.ACITM01-ACITM14" Name="AVAL" DataType="integer" Length="8">
  <Description>
    <TranslatedText xml:lang="en">Analysis Value</TranslatedText>
  </Description>
  <def:Origin Type="Derived"/>
</ItemDef>

<ItemDef OID="IT.ADQSADAS.AVAL.ACTOT" Name="AVAL" DataType="integer" Length="8">
  <Description>
     <TranslatedText xml:lang="en">Analysis Value</TranslatedText>
  </Description>
  <def:Origin Type="Derived"/>
</ItemDef>

<ItemDef OID="IT.ADQSADAS.PARAMCD" Name="PARAMCD" SASFieldName="APARAMCD"
  DataType="text" Length="8">
  <Description>
     <TranslatedText xml:lang="en">Parameter Code</TranslatedText>
  </Description>
  <CodeListRef CodeListOID="CL.PARAMCD_ADQSADAS"/>
  <def:Origin Type="Assigned"/>
</ItemDef>
```

4.6 Links to Supporting Documents

Two types of external documents, annotated case report forms (aCRFs) and supplemental documents, are often provided along with study datasets in a regulatory submission. Information about these documents is typically provided in the Define-XML document in order to facilitate the review of study data.

Some sponsors provide supplemental documentation in separate files, and those files are referenced from the Define-XML document. There are many reasons why additional or supplemental information is provided. Chief among them is the sponsor's desire to provide regulatory reviewers with further explanation or description of the datasets and their variables, or documentation that requires special formatting due to length or other considerations. For example, some clinical algorithms in a statistical analysis plan (SAP) may be complex, requiring additional explanation or formatting. These are best described in a flowchart or other graphic depiction. Comments may also be large enough to require special formatting, such as bullet points, for display purposes. Note, however, that users are encouraged to avoid duplication of documentation provided elsewhere in the Define-XML document.

- If an aCRF is provided, the Define-XML document should include a def:AnnotatedCRF element. The def:AnnotatedCRF element references def:leaf elements which identify the location of the external files relative to the folder where the Define-XML document is located. See Section 5.3.7, def:AnnotatedCRF Element and Section 5.13.16, def:leaf Element, respectively, for specification of these elements.
- If a supplemental document is provided, the Define-XML document should include a def:SupplementalDoc element. The def:SupplementalDoc element references def:leaf elements that identify the location of the external files relative to the folder where the Define- XML document is located. See Section 5.3.8, def:SupplementalDoc Element, for how to specify that element.

For SDTM data collected using a CRF, the variable definition must include a reference to pages in the aCRF to provide traceability information for reviewers. See Section 4.3.2, Origin/Source/Traceability Considerations, for business requirements, and Sections 5.3.12.3, def:Origin Element, and 5.3.7.1.1, def:PDFPageRef Element, for specification details.

4.6.1 Annotated CRF References

The def:AnnotatedCRF element contains a def:DocumentRef that references the annotated CRF document. The def:DocumentRef references a def:leaf element which, in turn, contains the xlink:href attribute that provides the location of the document. In this example, the aCRF is located in the same folder as the Define-XML document file and is named "acrf.pdf". For each ItemDef where the def:Origin Type attribute value is "Collected", the def:Origin will include a def:DocumentRef with the same def:leaf element reference.

Example 4.6.1.1 Annotated CRF Reference

```
<def:AnnotatedCRF>
   <def:DocumentRef leafID="LF.acrf"/>
</def:AnnotatedCRF>
...
<ItemDef OID="IT.DM.BRTHDTC" Name="BRTHDTC" DataType="date"
SASFieldName="BRTHDTC">
  <Description>
    <TranslatedText xml:lang="en">Date/Time of Birth</TranslatedText>
  </Description>
  <def:Origin Type="Collected" Source="Investigator">
    <def:DocumentRef leafID="LF.acrf">
      <def:PDFPageRef PageRefs="6" Type="PhysicalRef"/>
    </def:DocumentRef>
  </def:Origin>
</ItemDef>
...
<def:leaf ID="LF.acrf" xlink:href="acrf.pdf">
    <def:title>Annotated Case Report Form</def:title>
</def:leaf>
```

4.6.2 Supplemental Documentation References

The def:SupplementalDoc element contains 1 def:DocumentRef per supplemental document, each referencing a def:leaf element which in turn contains XLink information linking to the document.

Example 4.6.2.1 Supplemental Documentation Reference

```
<def:SupplementalDoc>
    <def:DocumentRef leafID="LF.ReviewersGuide"/>
    <def:DocumentRef leafID="LF.ComplexAlgorithms"/>
</def:SupplementalDoc>

<def:leaf ID="LF.ReviewersGuide" xlink:href="reviewersguide.pdf">
    <def:title>Reviewers Guide</def:title>
</def:leaf>

<def:leaf ID="LF.ComplexAlgorithms" xlink:href="complexalgorithms.pdf">
    <def:title>Complex Algorithms</def:title>
</def:leaf>
```

4.7 Computational Method Definitions

The MethodDef element (see Section 5.3.14, MethodDef Element) is intended to define the algorithm used to generate values for variables. The algorithm is linked to a variable or value using the MethodOID attribute in the corresponding ItemRef element (see Section 5.3.9.2, ItemRef Element).

To enhance traceability, users are encouraged to provide descriptions that include accurate and consistent references to source variables and derivations.

For cases where the algorithm description is longer than a few lines, the MethodDef can include a def:DocumentRef element to link to a section in a supplemental document containing the additional details (see Section 5.3.7.1, def:DocumentRef Element).

A MethodDef can also link to a text file that contains the specific programming code to be executed. However, the external location has to be agreed by sender and consumer.

Note: For regulatory submissions to the FDA, the specified locations have to conform to locations allowed in the Electronic Common Technical Document (eCTD) and the Study Data Specifications. See Section 2.2, References.

4.7.1 Short Method Definition

Example 4.7.1.1 MethodDef with Short Description

```
<MethodDef OID="MT.VSDY" Name="Algorithm to derive VSDY" Type="Computation">
    <Description>
        <TranslatedText xml:lang="en">VSDY = VSDTC-RFSTDTC+1 if VSDTC is on or
after RFSTDTC. VSDTC - RFSTDTC if VSDTC precedes RFSTDTC.</TranslatedText>
    </Description>
</MethodDef>
<MethodDef OID="MT.SESTDTC" Name="Algorithm to derive SESTDTC"
Type="Computation">
    <Description>
        <TranslatedText xml:lang="en">If Element = SCREEN, derived from SVSTDTC
where VISIT = SCREENING or from DS where DSDECOD = 'INFORMED CONSENT', whichever
is earliest. If Element = EOS, derived from DS where DSCAT = DISPOSITION EVENT.
```

```
For treatment Elements, derived from first EXSTDTC for the
element.</TranslatedText>
    </Description>
</MethodDef>
```

4.7.2 External Method Definition that References a Separate Document

Example 4.7.2.1 uses a Named Destination pointing to a specific place in a PDF document. Named Destinations have to be created within the PDF document to be able to hyperlink to them.

Example 4.7.2.1 MethodDef Documented in an External File

```
<!-- Method Definition: Algorithm included or expanded in an external file -->
<MethodDef OID="MT.EGDRVFL" Name="Algorithm to derive EGDRVFL"
Type="Computation">
    <Description>
        <TranslatedText xml:lang="en">EGDRVFL = "Y" for derived EGTESTCDs QTCB
and QTCF. Null otherwise. </TranslatedText>
    </Description>
    <def:DocumentRef leafID="LF.ComplexAlgorithms">
        <def:PDFPageRef PageRefs="EG" Type="NamedDestination"/>
    </def:DocumentRef>
</MethodDef>

<def:leaf ID="LF.ComplexAlgorithms" xlink:href="complexalgorithms.pdf">
    <def:title>Complex Algorithms</def:title>
</def:leaf>
```

4.7.3 Method Definition with Programming Code Reference

Example 4.7.3.1 links to a file with SAS programming code.

Note: At the time of publication of this document, the path in the xlink:href is not listed as applicable to SDTM in the context of a regulatory submission to the FDA. The example is provided only to illustrate the functionality.

Example 4.7.3.1 Method Definition with Programming Code Reference

```
<MethodDef OID="MT.QTCB" Name="Algorithm to derive QTCB" Type="Computation">
    <Description>
        <TranslatedText xml:lang="en">QTcB = QT interval / square root of (60 /
heart rate). For the complete algorithm see the referenced external
document.</TranslatedText>
    </Description>
    <def:DocumentRef leafID="LF.CODE.001"/>
</MethodDef>

<def:leaf ID="LF.CODE.001" xlink:href="../programs/QTCB-computation-sas.txt">
    <def:title>QTcB-Bazett's Correction Formula</def:title>
</def:leaf>
```

4.7.4 Method Definition with FormalExpression

Example 4.7.4.1 illustrates a Method definition that includes a FormalExpression in SAS programming code.

Example 4.7.4.1 Method Definition with FormalExpression Programming Code

```
<MethodDef OID="MT.USUBJID" Name="Algorithm to derive USUBJID"
Type="Computation">
    <Description>
        <TranslatedText xml:lang="en">Concatenation of STUDYID and
SUBJID</TranslatedText>
    </Description>
    <FormalExpression Context="SAS 9.0 or later, as part of a data step
assignment or proc sql select and update statements.">
        catx(".",STUDYID,SUBJID)
    </FormalExpression>
</MethodDef>
```

4.8 Comment Definitions

Define-XML allows Comments on all metadata definitions including the metadata version as a whole, dataset, variable, or value definitions. Comments are optional with respect to XML Schema validation; in the context of a regulatory submission, however, they provide a mechanism for sponsors to provide information that might help a reviewer.

Comments are also allowed on the Where Clause definitions for value-level metadata and, by convention, are recommended when a Where Clause references variables from different datasets.

The mechanism allows referencing short comments contained in the Define-XML document or long comments referenced in external documents. For comments in external documents, the reference can include specific pages.

Comments are not intended to replace a properly defined computational algorithm, which is expected for derived variables.

4.8.1 External Comment Definitions

Example 4.8.1.1 illustrates a comment definition referencing the section of the PDF file where complete documentation can be found, as well as a reference to the analysis program.

Example 4.8.1.1 Comment Referencing External File

```
<ItemGroupDef OID="IG.ADQSADAS" Name="ADQSADAS" SASDatasetName="ADQSADAS"
Repeating="Yes" IsReferenceData="No"
   Purpose="Analysis"  def:Structure="One record per subject per parameter per
analysis visit per analysis date"
   def:Class="BASIC DATA STRUCTURE"
def:CommentOID="COM.ADQSADAS"  def:ArchiveLocationID="LF.DOMAIN.ADQSADAS">
…
<!-- Comment Definition: Long Comment, included in a PDF file -->

<def:CommentDef OID="COM.ADQSADAS">
  <Description>
     <TranslatedText>See referenced dataset creation program and Analysis Data
Reviewer's Guide, Section 2.1</TranslatedText>
  </Description>
  <def:DocumentRef leafID="LF.ADQSADAS.PGM"/>
  <def:DocumentRef leafID="LF.ReviewersGuide ">
    <def:PDFPageRef PageRefs="Section2.1" Type="NamedDestination" Title="Section
2.1"/>
  </def:DocumentRef>
</def:CommentDef>

<def:leaf ID="LF.ReviewersGuide" xlink:href="ADRG.pdf">
  <def:title>Analysis Data Reviewer's Guide</def:title>
</def:leaf>

<def:leaf ID="LF.ADQSADAS.PGM" xlink:href="../programs/adqsadas-sas.txt">
  <def:title>adqsadas.sas</def:title>
</def:leaf>
```

4.8.2 Short Comment Definitions

Example 4.8.2.1 Short Comment Definition

```
<!-- Comments will be displayed in Stylesheet -->
<def:CommentDef OID="COM.DOMAIN.DM">
    <Description>
        <TranslatedText xml:lang="en">See Reviewer's Guide, Section 2.1
Demographics
        </TranslatedText>
    </Description>
</def:CommentDef>

<def:CommentDef OID="COM.DSDECOD">
    <Description>
        <TranslatedText xml:lang="en">CRF controlled terminology was mapped to
match CDISC controlled terminology.
        </TranslatedText>
    </Description>
</def:CommentDef>
```

4.9 Conditional Conformance Rules

Section 5, Specification, defines numerous Define-XML conformance rules. Many conformance rules are context-specific, including rules that apply when Define-XML is part of a regulatory submission. The following table lists the elements and attributes that are conditionally required, but mandatory in the context of a regulatory submission.

Define-XML Element or Attribute	Mandatory in the Submission Context Rule
ItemRef/@KeySequence	Required in the context of a regulatory submission for ItemRef elements corresponding to variables that make up the dataset key
ItemGroupDef/@Domain	Required in the context of a regulatory submission of SDTM or SEND datasets. Not applicable to ADaM datasets.
ItemGroupDef/@SASDatasetName	Required in the context of a regulatory submission when the data is submitted as SAS XPT files
ItemGroupDef/@ArchiveLocationID	Required in the context of a regulatory submission for each ItemGroupDef without a def:HasNoData attribute

ItemGroupDef/Description	Required in the context of a regulatory submission
ItemGroupDef/Alias	When used for regulatory submission, the Alias element is required for each ItemGroup that represents a supplemental qualifiers dataset.
ItemGroupDef/def:Class	Required when the ItemGroupDef describes a dataset that follows SDTM, SEND, or ADaM
ItemGroupDef/def:Class/def:SubClass	Required if the applicable IG defines the dataset described by the ItemGroupDef as requiring a Subclass.
CodeList/Alias CodeList/EnumeratedItem/Alias CodeList/CodeListItem/Alias	When used for regulatory submission, the Alias element is required for each CodeList that represents CDISC Controlled Terminology and for each EnumeratedItem or CodeListItem element that represents a defined term in CDISC Controlled Terminology.
ItemDef/@SASFieldName	Required in the context of a regulatory submission when data are submitted as SAS XPT files
ItemDef/def:Origin	For regulatory submissions, def:Origin metadata must be provided for all variables. It is at the sponsor's discretion whether to provide def:Origin at the variable or value level.

5 Specification

5.1 Define-XML Scope

A Define-XML document provides metadata for a set of datasets, variables, and associated information described in this document. This can be used for the regulatory submission of SDTM, ADaM, or SEND datasets for:

- A single clinical study
- A single nonclinical study
- An integrated summary of safety

5.2 Define-XML Structure

Because Define-XML is an extension to the CDISC ODM standard, Define-XML files follow the same basic structure as ODM files.

A Define-XML document includes the following key content components:

- XML header, the ODM root element, Study, and MetaDataVersion
- Standards definitions
- Information about linked PDF documents (e.g., annotated case report forms, Supplemental Data Definitions)
- Dataset definitions
- Variable definitions
- Value definitions (including Where Clause definitions)
- Controlled Terminology definitions
- Computational Method definitions
- Comment definitions
- External file Xlinks

The sections that follow describe what a Define-XML document can contain, and each element is described in the order in which it occurs in the XML. Elements that may be used in multiple contexts are presented where they first appear in the document.

Each section begins with a brief description of the element; this is followed by an element table and an attribute table. An element table describes the different aspects of an element's definition, whereas an attribute table describes the element's attributes. The following templates illustrate the layout of these tables, including headers and descriptions of the content.

In some cases, a section or subsection concludes with an XML example (accompanied by explanation of the example as necessary). However, most examples are provided in Section 4, General Specifications for Define-XML.

Note:

- The section hierarchy in this document does not reflect the XML structure. For example, the ODM and Study elements are described at the same level in this document, although the Study element in the XML is a child of the ODM element.
- The ODM elements AdminData, ClinicalData, ReferenceData, and Associations are not valid in Define-XML. The ODM ds:Signature element should not be used when Define-XML is used in a regulatory submission context.

Element Table Template

Element Name	Name of the element
Element XPath(s)	XPath showing where the element belongs in the XML
Element Textual Value	A description of the value of the element. If an element has no text value (e.g., it has child elements instead), then this cell is populated with "None".
Usage	• Requirement: This is populated with one of 3 values: ○ "Required" when at least 1 instance of the element is required ○ "Optional" when the element is optional ○ "Conditional" when at least 1 instance of the element is required under certain conditions. It will include the conditions under which the element is Required. • Cardinality: This indicates the number of instances expected (e.g. "Exactly one", "One or more") • Business Rule(s): Beyond what is required for schema validation, these represent conformance rules that must be satisfied for a Define-XML document to be considered compliant with this specification. • Other Information: This is populated with any other information about the element, including the conditions under which the element is included, how the schema is applied to support the model, relative position of the element in the model, and so on.
Attributes	A comma-delimited list of the attributes of this element. If the element has no attributes, this is populated with "None".
Child Elements	• A comma-delimited list of the immediate child elements of this element. If the element has no child elements, this is populated with "None". The order of child elements shown in the specification is the order in which they must appear in a Define-XML document. • A link to a child element will be provided when the child element is described in a different section of the document and not under a subsection of the element being described or in the section or subsection immediately following the current element.

Attribute Table Template

Attribute	Usage	Allowable Values	Description
Name of the attribute	• This is populated with "Required" when the attribute is required, "Optional" when the attribute is optional, or "Conditional" when the attribute is required under certain conditions. • It will include the conditions under which the attribute is Required. • Default: This will be populated with a default value if one is provided in the specification.	Any combination of the following: • Allowable Value: The only allowed value • Allowable Values: A comma-delimited list of the allowable values • Value Description: A textual description of allowable values • See Appendix *xx*: A reference to an appendix including a hyperlink to the appendix • Sample: An example	• A textual description of the attribute beyond what is included in the Allowable Values column. • Business Rule(s): Rules that have to be satisfied in addition to schema validation for a Define-XML document to be considered compliant with the Define-XML v2.1 specification

5.3 Define-XML Specification Details

5.3.1 XML Header

All XML document files must begin with an XML header, so the first line of the Define-XML document must be an XML header. The XML header indicates to applications that the remainder of the document is XML and specifies character encoding it uses.

The following example shows a Define-XML document using "UTF-8" character encoding.

Example 5.3.1.1 XML Document Header

```
<?xml version="1.0" encoding="UTF-8"?>
```

5.3.2 Stylesheet Reference

An XSL stylesheet can be referenced between the XML header and the ODM element, and allows the Define-XML document to be easily viewed in a web browser.

- If a stylesheet reference is provided, a browser can open the Define-XML document and display it according to the stylesheet. For a browser to correctly show the Define-XML, the referenced stylesheet must exist at the location specified. If a relative location is given for the stylesheet, it is relative to the location of the Define-XML document.
- The Define-XML standard does not dictate how a stylesheet should display a Define-XML document. An example stylesheet is provided; this can be altered to satisfy alternate visualization needs.
- If a stylesheet reference is not provided, a browser will display the XML contents of the Define-XML document.

The code in Example 5.3.2.1 references a stylesheet contained in the same location as the Define-XML document.

Example 5.3.2.1 Reference to an XSL Stylesheet

```
<?xml-stylesheet type="text/xsl" href="define2-x-x.xsl"?>
```

5.3.3 ODM Element

The first XML element in a file is known as the *root element*. For Define-XML files, the root element is the ODM element. This element identifies the namespaces used and includes attributes that affect the processing of the document as a whole. The namespaces illustrated in the code example are described in Section 3.1, File Conformity.

Element Name	ODM
Element XPath(s)	/ODM
Element Textual Value	None
Usage	• Requirement: Required • Cardinality: Exactly one • Other Information: This is the root element for the Define-XML document
Attributes	xsi:schemaLocation, ODMVersion, FileType, FileOID, CreationDateTime, AsOfDateTime, Originator, SourceSystem, SourceSystemVersion, def:Context
Child Elements	Study

Attribute	Usage	Allowable Values	Description
xsi:schemaLocation	Optional	Text Sample: "http://www.cdisc.org/ns/def/v2.1 define2-1-0.xsd"	Identifies the location of the schema for this XML document. First part is the Namespace URI; the second part is the location of the schema either on the Internet (e.g., http://www.abc.com/def.xsd) or on the local file system (e.g., def.xsd). Using a local copy of the schema rather than referencing a schema using a URL on the Web is recommended as it improves the probability that the software validating the Define-XML instance can find and access the appropriate files. However, when submitting a Define-XML to a regulatory authority, be aware that relative file references or references to a shared drive on a local area network may not work when the submission contents are transmitted to a different network location.
ODMVersion	Required	"1.3.2"	Identifies the ODM version that underlies the schema for the Define-XML document. ODMVersion is optional in the ODM standard, but required in Define-XML.
FileType	Required	"Snapshot"	Define-XML documents do not include audit trail elements, so the FileType is Snapshot.
FileOID	Required	Text	A unique identifier for this file. See the ODM specification for a discussion of FileOID recommendations.
CreationDateTime	Required	ISO 8601 datetime Sample: "2010-09-30T15:31:04"	The date and time when the specific version of the Define-XML document was created. This can be more accurately thought of as the "last modified" date and time.
AsOfDateTime	Optional	ISO 8601 datetime Sample: "2010-09-30T15:31:04"	The date and time at which the source database was queried to create this document
Originator	Optional	Text Sample: "Company XYZ"	Submission sponsor name
SourceSystem	Optional	Text	The name of the application that generated the Define-XML document
SourceSystem Version	Optional	Text	The version of the SourceSystem that generated the Define-XML document
def:Context	Required	"Submission" "Other"	Context in which the Define-XML document is used. This information can be used by a validator to perform checks conditional on the context.

Example 5.3.3.1 XML Header, Stylesheet Reference, and ODM Element

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="../Stylesheets/define2-x-x.xsl"?>
<ODM
    xmlns="http://www.cdisc.org/ns/odm/v1.3"
    xmlns:def="http://www.cdisc.org/ns/def/v2.1"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    ODMVersion="1.3.2"
```

```
     FileType="Snapshot"
     FileOID="Studycdisc01-Define2-XML_2.1.0/test"
     CreationDateTime="2017-02-13T19:05:32"
     Originator="CDISC Data Exchange Standards Team"
     def:Context="Submission">
```

5.3.4 Study Element

Study is the first element in the Define-XML document after the ODM element.

Element Name	Study
Element XPath(s)	/ODM/Study
Element Textual Value	*None*
Usage	• Requirement: Required • Cardinality: Exactly one • Other Information: The child element GlobalVariables contains child elements that capture high-level study information. The child element MetaDataVersion includes child elements to describe a collection of datasets.
Attributes	OID
Child Elements	GlobalVariables, MetaDataVersion

Attribute	Usage	Allowable Values	Description
OID	Required	Text	The unique ID of the Study. See ODM specification Section 2.11 for OID considerations.

5.3.4.1 GlobalVariables Element

GlobalVariables is the first child of Study.

Element Name	GlobalVariables
Element XPath(s)	/ODM/Study/GlobalVariables
Element Textual Value	None
Usage	• Requirement: Required • Cardinality: Exactly one
Attributes	None
Child Elements	StudyName, StudyDescription, ProtocolName

5.3.4.2 StudyName Element

StudyName is a name assigned to the study by the sponsor.

Element Name	StudyName
Element XPath(s)	/ODM/Study/GlobalVariables/StudyName
Element Textual Value	A short name assigned to the study by the sponsor

Usage	• Requirement: Required • Cardinality: Exactly one
Attributes	None
Child Elements	None

5.3.4.3 StudyDescription Element

StudyDescription is the second child of GlobalVariables.

Element Name	StudyDescription
Element XPath(s)	/ODM/Study/GlobalVariables/StudyDescription
Element Textual Value	A text description of the contents of the Study
Usage	• Requirement: Required • Cardinality: Exactly one • Other Information: Usually found in the high-level description of the study in the protocol document
Attributes	None
Child Elements	None

5.3.4.4 ProtocolName Element

ProtocolName is the name assigned to the study for external communication purposes.

Element Name	ProtocolName
Element XPath(s)	/ODM/Study/GlobalVariables/ProtocolName
Element Textual Value	A protocol name or number assigned to the study by a regulatory agency
Usage	• Requirement: Required • Cardinality: Exactly one • Other Information: Usually found in the protocol document or included in study master data file
Attributes	None
Child Elements	None

Example 5.3.4.4.1 Study and GlobalVariables Elements

```
<Study OID="cdisc01">
<GlobalVariables>
    <StudyName>NoMoreHeadaches</StudyName>
    <StudyDescription>Investigate the effectiveness of OurCure for
headaches.</StudyDescription>
    <ProtocolName>NCT00012345</ProtocolName>
</GlobalVariables>
```

5.3.5 MetaDataVersion Element

The MetaDataVersion element contains all the definitions related to the domains specified in the Define-XML. It includes attributes to identify the versions of the CDISC SDTM and Define-XML standards used by the submission and can also contain links to external documents that are frequently included such as the annotated case report form (aCRF) and Supplemental Data Definition document.

The tables below specify how the MetadataVersion element in a Define-XML document shall be constructed.

The XML in Example 5.3.5.1 indicates that this Define-XML instance follows Version 2.1 of the Define-XML specification. Content standards are described using the /ODM/Study/MetaDataVersion/Standards/def:Standards element; for examples, see Section 4.1.2, Examples of Standards Metadata in Define-XML.

Element Name	MetaDataVersion
Element XPath(s)	/ODM/Study/MetaDataVersion
Element Textual Value	None
Usage	• Requirement: Required • Cardinality: One
Attributes	OID, Name, Description, def:DefineVersion
Child Elements	def:Standards, def:AnnotatedCRF, def:SupplementalDoc, def:ValueListDef, def:WhereClauseDef, ItemGroupDef, ItemDef, CodeList, MethodDef, def:CommentDef, def:leaf

Attribute	**Usage**	**Allowable Values**	**Description**
OID	Required	Text	Unique ID for the MetaDataVersion. See ODM specification Section 2.11 for OID considerations.
Name	Required	Text	Name for the MetaDataVersion
Description	Optional	Text Sample: "Study CDISC01 Data Definitions"	Description of the MetaDataVersion
def:DefineVersion	Required	"2.1.n"	Version of Define-XML to which the file conforms. The *n* in "2.1.n" reflects updates to the schema.
def:CommentOID	Optional	Text	• Reference to the unique ID of a def:CommentDef element that contains the comment for the MetaDataVersion element. See Section 4.8, Comment Definitions. • Business Rule: Must match the OID of a def:CommentDef element in the same MetaDataVersion.

Example 5.3.5.1 MetaDataVersion Element

```
<MetaDataVersion OID="CDISC01.SDTMIG.3.1.2.SDTM.1.2"
    Name="Study CDISC01, Data Definitions" Description="Study CDISC01, Data
Definitions"
    def:DefineVersion="2.1.0">
```

5.3.6 def:Standards Element

The def:Standards element provides a container for the list of Standard elements.

Element Name	def:Standards
Element XPath(s)	/ODM/Study/MetaDataVersion/def:Standards
Element Textual Value	None
Usage	• Requirement: Conditional. Required when def: Context="Submission" but optional otherwise. • Cardinality: Exactly One • Other Information: This is the container for child Standard element(s) referenced in the MetaDataVersion for the Study.
Attributes	None
Child Elements	def:Standard

5.3.6.1 def:Standard Element

The def:Standard element describes each standard used within the MetaDataVersion element.

Element Name	def:Standard
Element XPath(s)	/ODM/Study/MetaDataVersion/def:Standards/def:Standard
Element Textual Value	None
Usage	• Requirement: Required (when parent element def:Standards is present) • Cardinality: One or more
Attributes	OID, Name, Type, PublishingSet, Version, def:CommentOID
Child Elements	None

Attribute	**Usage**	**Allowable Values**	**Description**
OID	Required	Text	
Name	Required	Text • The list of values is conditional to each value of the def:Standard/@Type attribute. The list of allowable values for each def:Standard/@Type value is provided by an extensible codelist in the CDISC Controlled Terminology. • The name and concept code for this codelist are TBD.	• Name of an external standard to which the data conforms. • See Section 4.1.1, Standards Considerations.
Type	Required	Text • The list of allowable values is provided by a codelist in the CDISC Controlled Terminology and is extensible. • The name and concept code for this codelist are TBD.	• The type of standard. • See Section 4.1.1, Standards Considerations.
PublishingSet	Conditionally required when Type="CT"	Text • The list of allowable values is provided by a codelist in the CDISC Controlled Terminology and is extensible.	• The name of the publishing set that contains the published standard. • See Section 4.1.1, Standards Considerations.

	Not applicable for other standard types	- The name and concept code for this codelist are TBD.	
Version	Required	Text - The version identifier should match the version identifier used in the standard publication.	- Version of an external standard to which the data conforms. - See Section 4.1.1, Standards Considerations. See also Section 1, Introduction, for a list of the versions of CDISC standards supported by Define-XML v2.1.0 at the time of publication.
Status	Optional	Text - The list of allowable values is provided by a codelist in the CDISC Controlled Terminology and is not extensible. - The name and concept code for this codelist are TBD.	Publishing status of the standard. If no value is provided, the assumption is that the publishing status is Final.
def:CommentOID	Optional	Text	- Reference to the unique ID of a def:CommentDef element that contains the comment for the def:Standard element - See Section 4.8, Comment Definitions. - Business Rule: Must match the OID of a def:CommentDef in the same MetaDataVersion

5.3.7 def:AnnotatedCRF Element

An annotated CRF is a PDF document that provides the mapping of data collection fields to variables or to discrete variable values contained within SDTM datasets.

Element Name	def:AnnotatedCRF
Element XPath(s)	/ODM/Study/MetaDataVersion/def:AnnotatedCRF
Element Textual Value	None
Usage	- Requirement: Optional - Cardinality: Zero or One - Other Information: Contains the DocumentRef for the annotated CRF
Attributes	None
Child Elements	def:DocumentRef

5.3.7.1 def:DocumentRef Element

The def:DocumentRef element references a def:leaf element that, in turn, provides information about the document file.

Element Name	def:DocumentRef
Element XPath(s)	- /ODM/Study/MetaDataVersion/def:AnnotatedCRF/def:DocumentRef - /ODM/Study/MetaDataVersion/def:SupplementalDoc/def:DocumentRef - /ODM/Study/MetaDataVersion/ItemDef/def:Origin/def:DocumentRef

	• /ODM/Study/MetaDataVersion/MethodDef/def:DocumentRef • /ODM/Study/MetaDataVersion/def:CommentDef/def:DocumentRef
Element Textual Value	None
Usage	• Requirement: Conditional ○ Required for def:AnnotatedCRF and def:SupplementalDoc ○ Optional for def:Origin, MethodDef, and def:CommentDef • Cardinality: Zero or more Business Rule: If multiple documents are provided, there will be multiple def:DocumentRef child elements within the def:SupplementalDoc element.
Attributes	leafID
Child Elements	def:PDFPageRef

Attribute	Usage	Allowable Values	Description
leafID	Required	Text	Reference to the unique ID of the def:leaf element that contains the location of a file containing a document.

5.3.7.1.1 def:PDFPageRef Element

This element is the container for PDF page references.

Element Name	def:PDFPageRef
Element XPath(s)	• /ODM/Study/MetaDataVersion/ItemDef/def:Origin/def:DocumentRef/def:PDFPageRef • /ODM/Study/MetaDataVersion/MethodDef/def:DocumentRef/def:PDFPageRef • /ODM/Study/MetaDataVersion/def:CommentDef/def:DocumentRef/def:PDFPageRef
Element Textual Value	None
Usage	• Requirement: Conditional ○ Required for def:Origin/@Type="Collected" and def:Origin/@Source in ("Investigator","Subject") ○ Optional in all other cases • Cardinality: Zero or more Note that when the Origin type is Collected, there is an assumption that an annotated image of the entry page is available.
Attributes	Type, PageRefs, FirstPage, LastPage, Title
Child Elements	None

Attribute	Usage	Allowable Values	Description
Type	Required	Allowable Values: PhysicalRef, NamedDestination	• Type of page for page references indicated in the PageRefs attribute • Business Rule: When Type="NamedDestination", NamedDestinations have to be created within the PDF document to be able to link to them via a hyperlink.
PageRefs	Optional	Text Sample: "17 20 32"	List of PDF pages, separated by spaces

FirstPage	Conditional Required if PageRefs is not provided	Integer	First page in a range of pages. Note that the way to indicate the range of pages depends on the associated Type attribute provided.
LastPage	Conditional Required if PageRefs is not provided	Integer	Last page in a range of pages. Note that the way to indicate the range of pages depends on the associated Type attribute provided.
Title	Optional	Text	Alternative label to provide a more specific and descriptive reference to a page link

5.3.8 def:SupplementalDoc Element

The def:SupplementalDoc element contains a DocumentRef for each supplemental document. Typically for a SDTM submission, there will be a reference to the Study Data Reviewers Guide; for an ADaM submission, there will be a reference to the Analysis Data Reviewers' Guide.

Element Name	def:SupplementalDoc
Element XPath(s)	/ODM/Study/MetaDataVersion/def:SupplementalDoc
Element Textual Value	None
Usage	• Requirement: Optional • Cardinality: Zero or One • Business Rule: If multiple documents are provided, there will be multiple def:DocumentRef child elements within the def:SupplementalDoc element.
Attributes	None
Child Elements	def:DocumentRef

5.3.9 def:ValueListDef Element

The table below specifies the XML structure for valuelist metadata.

Element Name	def:ValueListDef
Element XPath(s)	/ODM/Study/MetaDataVersion/def:ValueListDef
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: Required for each unique value of the ValueListOID attribute within the MetaDataVersion • Business Rule: For SDTM SUPPQUAL datasets, a def:ValueListDef element must be provided to describe the QVAL variable. • Other Information: Contains ItemRef elements that reference ItemDef elements that provide the value-level metadata details
Attributes	OID
Child Elements	Description, ItemRef

Attribute	Usage	Allowable Values	Description
OID	Required	Text	Unique ID for the Value List See ODM specification Section 2.11 for OID considerations.

5.3.9.1 Description Element

The Description element is used to provide a readable summary of the parent element contents.

Element Name	Description
Element XPath(s)	• /ODM/Study/MetaDataVersion/def:ValueListDef/Description • /ODM/Study/MetaDataVersion/ItemGroupDef/Description • /ODM/Study/MetaDataVersion/ItemDef/Description • /ODM/Study/MetaDataVersion/ItemDef/def:Origin/Description • /ODM/Study/MetaDataVersion/CodeList/Description • /ODM/Study/MetaDataVersion/CodeList/CodeListItem/Description • /ODM/Study/MetaDataVersion/CodeList/EnumeratedItem/Description • /ODM/Study/MetaDataVersion/MethodDef/Description • /ODM/Study/MetaDataVersion/def:CommentDef/Description
Element Textual Value	None
Usage	• Requirement: Conditional ○ Required for all ItemGroupDef and for ItemDef elements corresponding to Variable definitions in the context of a regulatory submission ○ Required for ItemDef elements for value level metadata for supplemental qualifiers and for non-standard variables in the context of a regulatory submission ○ Required for MethodDef/Description and def:CommentDef/Description ○ Optional for def:ValueListDef/Description, CodeList/Description, CodeList/CodeListItem/Description, CodeList/EnumeratedItem/Description, and def:Origin/Description • Cardinality: Zero or One • Business Rules: ○ For SDTM or SEND standard domains, the ItemGroupDef/Description and ItemDef/Description should exactly match the label for the relevant standard dataset or variable. ○ For ADaM datasets, the ItemGroupDef/Description and ItemDef/Description should adhere to the naming rules for labels specified in the ADaMIG and should exactly match the label of the respective dataset or variable. ○ For custom domains or datasets, sponsors should provide a short description of the type of data contained within the dataset or variable in ItemGroupDef/Description and ItemDef/Description. ○ Conventions described in IG documents for the specific standard should be followed.
Attributes	None
Child Elements	TranslatedText

5.3.9.1.1 TranslatedText Element

Element Name	TranslatedText
Element XPath(s)	• /ODM/Study/MetaDataVersion/def:ValueListDef/Description/TranslatedText • /ODM/Study/MetaDataVersion/ItemGroupDef/Description/TranslatedText • /ODM/Study/MetaDataVersion/ItemDef/Description/TranslatedText • /ODM/Study/MetaDataVersion/ItemDef/def:Origin/Description/TranslatedText

	• /ODM/Study/MetaDataVersion/CodeList/CodeListItem/Decode/TranslatedText • /ODM/Study/MetaDataVersion/CodeList/Description/TranslatedText • /ODM/Study/MetaDataVersion/CodeList/CodeListItem/Description/TranslatedText • /ODM/Study/MetaDataVersion/CodeList/EnumeratedItem/Description/TranslatedText • /ODM/Study/MetaDataVersion/MethodDef/Description/TranslatedText • /ODM/Study/MetaDataVersion/def:CommentDef/Description/TranslatedText
Element Textual Value	Text string
Usage	• Requirement: Required • Cardinality: One or more. Multiple TranslatedText child elements can be used to provide the dataset description in different languages, preferably one for each language. • Business Rules: ○ For submissions to the FDA, text content must be in English. ○ In cases where SAS Transport files are the transport format, the value provided in the ItemGroupDef element should match the dataset label provided in the SAS Transport file.
Attributes	xml:lang
Child Elements	None

Attribute	Usage	Allowable Values	Description
xml:lang	Optional Default: "en", but depends on locale	See IETF Trust, *Tags for Identifying Languages* (Section 2.2, References) Samples: • "en" for English • "en-GB" for British English	Code representing the language of the enclosed text value Business Rules: • The schema requires xml:lang to be unique within parent element. • If only one TranslatedText element is provided, the xml:lang is optional.

5.3.9.2 ItemRef Element

For each Variable within a dataset, the ItemGroupDef must include an ItemRef element. ItemRef elements are also included in ValueListDef elements.

Element Name	ItemRef
Element XPath(s)	• /ODM/Study/MetaDataVersion/ItemGroupDef/ItemRef • /ODM/Study/MetaDataVersion/def:ValueListDef/ItemRef
Element Textual Value	None
Usage	• Requirement: Required • Cardinality: ○ One for each dataset variable (when parent node is an ItemGroupDef element) ○ One for each distinct value to be defined (when parent node is a ValueListDef element)
Attributes	ItemOID, OrderNumber, Mandatory, KeySequence, Role, RoleCodeListOID, MethodOID, def:IsNonStandard, def:HasNoData
Child Elements	def:WhereClauseRef (valid only when parent node is a ValueListDef)

Attribute	Usage	Allowable Values	Description
ItemOID	Required	Text	• Reference to the unique ID of an ItemDef element • See ODM specification Section 2.11 for OID considerations. • Business Rule: Each ItemOID used in an ItemRef must match an OID for an ItemDef.
OrderNumber	Optional	Integer	Display order of the variable within the domain dataset Business Rules: • See Section 4.1.4 of the SDTMIG for variable ordering requirements for the SDTM and ADaMIG Section 3 for information about variable ordering for ADaM datasets. • If OrderNumber is not provided, displays should list items in the order the ItemRef elements appear within the ItemGroup or ValueList. • If the OrderNumber is provided, displays should list items in the order given by the OrderNumber within the ItemGroup or ValueList.
Mandatory	Required	"Yes", "No"	Business Rules: • For SDTM-based variables where Core is "Req" (Required), the value of Mandatory should be set to "Yes". • For variables where Core is "Exp" (Expected) or "Perm" (Permissible) and the sponsor does not require a more restrictive condition, Mandatory should be set to "No". • Variables where Mandatory="Yes" must not have a null value.
KeySequence	Conditional Each variable that is part of the dataset key must have a KeySequence attribute. In the context of a regulatory submission, a key must be provided for each data set (ODM/@def:Context ="Submission").	Integer	Including the KeySequence attribute for an ItemRef corresponding to a dataset variable indicates that the variable is a key for the dataset defined by the enclosing ItemGroupDef. The numeric value indicates the variable position within the key.
MethodOID	Conditional This attribute and the associated MethodDef are Required when the Type attribute for the def:Origin child element of the referenced ItemDef is "Derived".	Text	Reference to the unique ID of a MethodDef element

	Otherwise, this attribute is Optional.		
Role	Optional	Text Allowable Values: • For SDTM or SEND datasets, any valid SDTM or SEND role as defined in the corresponding model or IG. • Not applicable for ADaM • For other datasets, any value can be used.	Variable Role defines how the variable defined by the corresponding ItemDef element is used within the dataset. Notes: • Variable Role values provided by the SDTMIG or SENDIG are used for standard variables in standard domains. • Variable Role values for standard variables not defined in an IG are defined in the SDTM. • At the time of publication of this document, role is not defined for ADaM variables or parameters.
RoleCodeListOID	Optional	Text	Reference to the unique ID of a CodeList element that defines the values for the codes given in the Role attribute
def:IsNonStandard	Optional Permitted for any ItemRef that represents a non-standard variable in an SDTM or SEND dataset when the ODM Context attribute value is "Other" Is not used with an ItemRef that represents a value level metadata for an SDTMIG SUPPQUAL dataset Has no meaning for ADaM, so shall not be used for ADaM variables	"Yes"	A non-standard variable is one that is not defined in the SDTMIG or the underlying SDTM model of a standard referenced within the Define-XML document. See Section 4.1, Standards References.
def:HasNoData	Conditional Required in the context of a regulatory submission when the dataset variable defined in the associated ItemDef has no data values (ODM@def:Context ="Submission")	"Yes"	• Used to indicate that an ItemRef that represent a dataset's variable has no data. Note that *variables* refer to both standard and non-standard/ supplemental qualifiers variables (/ODM/Study/MetaDataVersion/ItemGroupDef/ItemRef or /ODM/Study/MetaDataVersion/def:ValueListDef/ItemRef). • Business Rule: A comment must be included to explain why no data is present for dataset's variables that were planned for use in the study.

5.3.9.2.1 def:WhereClauseRef Element

The def:WhereClauseRef references the WhereClauseDef element that describes the conditions under which the variable values are defined by the referenced ItemDef.

If more than 1 is provided, the parent def:ValueList Def applies for multiple slices.

Element Name	def:WhereClauseRef
Element XPath(s)	/ODM/Study/MetaDataVersion/def:ValueListDef/ItemRef/def:WhereClauseRef
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: One or more def:WhereClauseRef elements is Required for each ItemRef child element within a def:ValueListDef. • Business Rules: o Not allowed as a child element of an ItemRef element if the parent node is a def:ItemGroupDef element. It will be considered non-conforming. o If more than one def:WhereClauseRef elements appear in an ItemRef child element within a def:ValuelistDef, they are interpreted as a logical OR.
Attributes	WhereClauseOID
Child Elements	None

Attribute	Usage	Allowable Values	Description
WhereClauseOID	Required	Text	Reference to the unique ID of a def:WhereClauseDef element

5.3.10 def:WhereClauseDef Element

The def:WhereClausDef element specifies a condition.

Element Name	def:WhereClauseDef
Element XPath(s)	/ODM/Study/MetaDataVersion/def:WhereClauseDef
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: A def:WhereClause is required for each unique value of the WhereClauseOID attribute value in a def:WhereClauseRef element within the MetaDataVersion.
Attributes	OID, def:CommentOID
Child Elements	RangeCheck

Attribute	Usage	Allowable Values	Description
OID	Required	Text	Unique ID for the WhereClauseDef. See ODM specification Section 2.11 for OID considerations.
def:CommentOID	Conditional Required when RangeCheck includes def:ItemOID values that belong to different ItemGroupDef elements	Text	Reference to the unique ID of a def:CommentDef that describes how to join the datasets when the WhereClause includes references to variables in different datasets.

5.3.10.1 RangeCheck Element

Element Name	RangeCheck
Element XPath	/ODM/Study/MetaDataVersion/def:WhereClauseDef/RangeCheck
Element Textual Value	Contains the comparison specification defining the WhereClause condition
Usage	• Requirement: Required • Cardinality: Each def:WhereClauseDef element must have at least one RangeCheck child element. • Other Information: If multiple RangeChecks are given, a dataset row must satisfy all the RangeChecks.

Attribute	**Usage**	**Allowable Values**	**Description**
Comparator	Required	"LT","LE","GT","GE","EQ", "NE","IN","NOTIN"	Comparison operator for WhereClause
SoftHard	Required	"Soft","Hard"	• If an actual data value fails the constraint, it is either rejected (Hard constraint) or a warning is produced (Soft constraint). • Business Rule: The SoftHard attribute has no meaning in the Define-XML context. Although ODM requires a value equal to "Hard" or "Soft", neither value implies any meaning to the enclosing RangeCheck or WhereClauseDef element.
def:ItemOID	Required	Text	Reference to the unique ID of an ItemDef that is used to compare with the CheckValue

5.3.10.2 CheckValue Element

Element Name	CheckValue
Element XPath(s)	/ODM/Study/MetaDataVersion/def:WhereClauseDef/RangeCheck/CheckValue
Element Textual Value	The comparison value
Usage	• Requirement: Required • Cardinality: One or More
Attributes	None
Child Elements	None

5.3.11 ItemGroupDef Element

The ItemGroupDef element is used to describe the dataset metadata in XML.

Element Name	ItemGroupDef
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemGroupDef
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: One or more. • Business Rule: In the context of a regulatory submission, the regulatory technical conformance guides and the implementation guides for each relevant standard specify the datasets required.
Attributes	OID, Name, Domain, Purpose, SASDatasetName, Repeating, IsReferenceData, def:Structure, def:ArchiveLocationID, def:CommentOID, def:IsNonStandard, def:StandardOID, def:HasNoData
Child Elements	Description, ItemRef, Alias, def:Class, def:leaf

Attribute	**Usage**	**Allowable Values**	**Description**
OID	Required	Text	Unique ID for the ItemGroupDef (dataset). See ODM specification Section 2.11 for OID considerations.
Name	Required	Text	Short description for the ItemGroup. Note that the Name attribute must be the same as SASDatasetName if SAS is being used as a transport mechanism. If the transport mechanism is not SAS-based, this attribute will contain the name of the dataset.
Domain	Conditional Required for SDTM and SEND datasets except for RELREC and POOLDEF; not applicable for ADaM	Text	Domain as specified in the SDTMIG or SENDIG Business Rules: • For supplemental qualifiers datasets, Domain must contain the Domain name of the parent dataset. • For split domains, Domain must contain the Domain name of the parent dataset.
Purpose	Required	"Tabulation", "Analysis"	Purpose of domain or dataset Business Rules: • For SDTM and SEND, use Tabulation • For ADaM, use Analysis
SASDatasetName	Conditional Required in the context of a regulatory submission	Text	Root name of SAS dataset contained in the SAS Transport file containing the ItemGroup data. The root name is file name in upper case without the ".xpt" extension. For split datasets, the value of the SASDatasetName attribute and the file name referenced by the def:ArchiveLocationID attribute will contain a suffix component to identify the individual split file (e.g., QSCG, QSCS and qscg.xpt, qscs.xpt). Business Rules: • Must conform to SAS Transport file naming rules

			• If a value is provided, it should be in upper case .
Repeating	Required	"Yes", "No"	• Set to "Yes" if the dataset may contain more than 1 record per subject or pool; set to "No" if the dataset may contain only 1 record per subject or pool • Business Rule: When IsReferenceData="Yes", set Repeating="No".
IsReferenceData	Optional Default Value: No	"Yes", "No" Sample: • For Trial Design domains: Yes • For DM domain: No	Indicates whether the dataset contains reference data (not subject data)
def:Structure	Required	Text Samples: • MH domain: "1 record per medical history event per subject" • VS domain: "1 record per vital sign measurement per visit per subject" • ADQSADAS dataset: "1 record per subject per parameter per analysis visit per analysis date"	Description of the level of detail represented by individual records in the dataset
def:ArchiveLocationID	Conditional Required in the context of a regulatory submission for each ItemGroupDef that does not include def:HasNoData="Yes"	Text	Reference to the unique ID of a def:leaf that provides the actual location and file name of the SAS transport file • If provided, this should match the leaf:id attribute of the def:leaf child element • If not provided, the root dataset file name is expected to be the same as the Name attribute.
def:CommentOID	Optional	Text	Reference to the unique ID of a def:CommentDef element that contains the comment for the ItemGroupDef (dataset). See Section 4.8, Comment Definitions. • For ADaM datasets, the comment referenced by this value is the dataset Documentation metadata. Business Rules: • Must match the OID of a def:CommentDef element in the same MetaDataVersion • If def:HasNoData attribute is provided, a def:CommentOID attribute must be populated and the corresponding def:CommentDef element must be provided.

def:IsNonStandard	Conditional Required for ADaM, SDTM, or SEND if def:StandardOID is not provided	"Yes"	Should not be provided when def:StandardOID is provided. See Section 4.1, Standards References.
def:StandardOID	Conditional Each ItemGroupDef for an ADaM, SDTM, or SEND dataset must provide either a def:StandardOID or set def:IsNonStandard="Yes".	Text	• References the OID of a def:Standard element of Type="IG" • Business Rule: StandardOID must match the OID for a def:Standard element with Type="IG".
def:HasNoData	Optional	"Yes"	• Used to indicate that a ItemGroupDef has no data. May be used at sponsor's discretion or if required by a regulatory authority. • Business Rule: A comment must be included to explain why no data are present for datasets that were planned for use in the study.

5.3.11.1 Alias Element

The Alias element is used to provide

- the domain description in the case where an ItemGroupDef represents a supplemental qualifiers dataset or a dataset from a split domain;
- SAS variable or dataset names longer than 8 characters
- a reference to CDISC Controlled Terminology (e.g., the C-Code when a CodeList, CodeListItem, or EnumeratedItem corresponds to a CDISC Controlled Terminology).

Element Name	Alias
Element XPath(s)	• /ODM/Study/MetaDataVersion/ItemGroupDef/Alias • /ODM/Study/MetaDataVersion/ItemDef/Alias • /ODM/Study/MetaDataVersion/CodeList/Alias • /ODM/Study/MetaDataVersion/CodeList/CodeListItem/Alias • /ODM/Study/MetaDataVersion/CodeList/EnumeratedItem/Alias
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: Zero or more ○ For SDTM based datasets, the Alias element is required for each ItemGroupDef that represents a split dataset or a supplemental qualifiers dataset. ○ When used for regulatory submission of SDTM, ADaM, or SEND metadata, the Alias element is required for each CodeList referenced by an ItemDef for variables that require CDISC Controlled Terminology. Each EnumeratedItem of CodeListItem element representing a CDISC Controlled Terminology-defined term also requires an Alias element.
Attributes	Context, Name
Child Elements	None

Attribute	Usage	Allowable Values	Description
Context	Required	Text The allowable values are not restricted by the schema. The following Context values are noted within this specification: • As a child element of an ItemGroupDef element: DomainDescription, SAS • As a child element of an ItemDef element: SAS • As a child element of a CodeList element: nci:ExtCodeID, SAS • As a child element of a CodeListItem or EnumeratedItem element: nci:ExtCodeID Note: In the context of a regulatory submission, the DomainDescription is not used.	Indicates the context or setting where the Alias Name attribute applies
Name	Required	Text The allowable values are not restricted by the schema. The following Context values are noted within this specification: • As a child element of an ItemGroupDef element when Context="DomainDescription": description of the domain provided in /ODM/Study/MetaDataVersion/ItemGroupDef@Domain • As a child element of an ItemGroupDef element when Context="SAS": the SAS dataset name • As a child element of an ItemDef element when Context="SAS": the SAS variable name • As a child element of a CodeList element when Context="nci:ExtCodeID": C-Code for corresponding CDISC Controlled Terminology codelist. • As a child element of a CodeListItem or EnumeratedItem, when Context="nci:ExtCodeID": C-Code for corresponding CDISC Controlled Terminology term	Alternative Name or reference for parent element

5.3.11.2 def:Class Element

For each dataset definition that follows the SDTM or ADaM, the def:Class element shall be provided to identify which predefined Class within the model applies to the definition.

Element Name	def:Class
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemGroupDef/def:Class
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: One def:Class element is allowed for each ItemGroupDef element for a dataset that follows the CDISC SDTMIG, SENDIG, or ADaMIG. • Business Rule: For analysis datasets, if the ItemGroupDef def:IsNonStandard attribute is used, the def:Class should not be provided.
Attributes	Name
Child Elements	def:SubClass

Attribute	Usage	Allowable Value	Description
Name	Required	Text	Text must follow CDISC Controlled Terminology for General Observation Class.

5.3.11.2.1 def:SubClass Element

For dataset definitions that follow a model where SubClasses have been defined, the def:SubClass child element of the def:Class element identifies the specific SubClass that applies to the definition in the parent ItemGroupDef element. It is possible to have multi-level/nested SubClasses.

Element Name	def:SubClass
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemGroupDef/def:Class/def:SubClass
Element Textual Value	None
Usage	• Requirement: Optional • Cardinality: Defined by the applicable model IG.
Attributes	Name, ParentClass
Child Elements	None

Attribute	Usage	Allowable Value	Description
Name	Required	Text	Text must follow CDISC Controlled Terminology for SubClass.
ParentClass	Optional	Text	Text must follow CDISC Controlled Terminology for Class or SubClass. For a nested or multi-level SubClass, the ParentClass attribute is used to define the hierarchy.

5.3.12 ItemDef Element

The ItemDef element is used to represent variable or value-level metadata. Value-level metadata is always a specialization of variable metadata. Use one ItemDef element per unique variable or value.

Element Name	ItemDef
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemDef
Element Textual Value	None
Usage	• Requirement: Required • Cardinality: An ItemDef element is required for each ItemOID value that appears in an ItemRef contained in a MetaDataVersion.
Attributes	OID, Name, DataType, Length, SignificantDigits, SASFieldName, def:DisplayFormat, def:CommentOID
Child Elements	Description, CodeListRef, def:Origin, def:ValueListRef, Alias

Attribute	Usage	Allowable Values	Description
OID	Required	Text	Unique ID for the ItemDef (variable/value). See ODM specification Section 2.11 for OID considerations.

Name	Required	Text	Dataset Variable name or Variable Value Name
DataType	Required	See Section 4.3.1, Data Type Considerations, for a list of valid Define-XML DataType values. Samples: For SDTM, SEND, and ADaM, variables one of "Text", "Float", "Integer", "Date", "Datetime"	The data type of the variable or value
Length	Conditional Required if DataType is "Text", "Integer", or "Float"	Integer	The variable length in characters. Business Rules: • Length is the maximum expected variable length. It is not based on the length of the data collected. • Length should be present only when DataType is "Text", "Integer", or "Float". • For ItemDef elements referenced from a ValueListDef, the Length must be less than or equal to the length of the parent variable.
SignificantDigits	Conditional Required if DataType is "Float"	Integer	• The number of digits following the decimal point in a floating point number • Business Rule: When DataType is float, both Length and SignificantDigits must be provided.
SASFieldName	Optional	Text	• SAS Variable Name • Business Rule: Follow rules for Variable names in SAS Transport files.
def:DisplayFormat	Optional	Text Samples: • 8.2 (means display floating point variable values to the second decimal place) • SAS date9 (means display a numeric ADaM date variable in the format DDMMMYYYY)	Display format supports data visualization of numeric float and date values. In the context of a regulatory submission, the def:DisplayFormat is expected to be a SAS display format.
def:CommentOID	Optional	Text	• Reference to the unique ID of a def:CommentDef element that contains the comment for the ItemDef (variable). See Section 4.8, Comment Definitions. • Business Rule: Must match the OID of a def:CommentDef element in the same MetaDataVersion

5.3.12.1 CodeListRef Element

Provides a reference to the Codelist element defining controlled terminology if applicable.

Element Name	CodeListRef
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemDef/CodeListRef
Element Textual Value	None
Usage	• Requirement: Optional • Cardinality: One

	• Business Rule: If a variable or value definition includes CDISC Controlled Terminology, a CodeList element should be provided as a child element on the ItemDef.
Attributes	CodeListOID
Child Elements	None

Attribute	Usage	Allowable Values	Description
CodeListOID	Required	Text	• Reference to the unique ID of a CodeList element that defines controlled terminology for the variable or values defined by the ItemDef. See ODM specification Section 2.11 for OID considerations. • Business Rule: Must match the OID of a CodeList element in the same MetaDataVersion.

5.3.12.2 def:ValueListRef Element

The def:ValueListRef element is the OID of the def:ValueListDef that contains the valuelist definition associated with the variable. If value-level metadata is required for a variable, a def:ValueListRef element should be provided as a child element on the ItemDef for the variable definition.

Element Name	def:ValueListRef
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemDef/def:ValueListRef
Element Textual Value	None
Usage	• Requirement: Optional • Cardinality: One
Attributes	ValueListOID
Child Elements	None

Attribute	Usage	Allowable Values	Description
ValueListOID	Required	Text	• Reference to the unique ID of a def:ValueListDef element that provides value-level metadata. See ODM specification Section 2.11 for OID considerations. • Business Rule: Must match the OID of a def:ValueListDef in the same MetaDataVersion.

5.3.12.3 def:Origin Element

The def:Origin element is intended to be define the Origin metadata for variables in SDTM or SEND datasets or source metadata for ADaM datasets. The use of the ODM Origin attribute within Define-XML has been deprecated and is not valid in Define-XML v2.1.

Element Name	def:Origin
Element XPath(s)	/ODM/Study/MetaDataVersion/ItemDef/def:Origin
Element Textual Value	None
Usage	• Requirement: Conditional o For regulatory submissions, def:Origin metadata must be provided for all SDTM, ADaM, or SEND variables. o In some cases, def:Origin will be provided only at the Value level. • Cardinality: Zero or more

	○ Multiple def:Origins may be provided when there are multiple sources for a single Variable but no way to construct a ValueList that clearly identifies the cases where each def:Origin applies. • Business Rules: ○ If all data rows for the variable represented by the parent ItemDef have the same origin, def:Origin should be provided at the variable level. ○ If the parent ItemDef includes a def:ValueListRef and each ItemDef element referenced in the corresponding def:ValueListDef includes a def:Origin element, there should be no def:Origin on the parent ItemDef. ○ If the variable or value is derived, the corresponding ItemRef must include a MethodOID attribute that references the corresponding MethodDef. ○ When def:Origin/@Type="Collected" and def:Origin/@Source in ("Investigator", "Subject"), there must be a def:DocumentRef child element; in addition, def:DocumentRef/@leafID must match the ID attribute of a def:leaf element corresponding to the annotated CRF document.
Attributes	Type, Source
Child Elements	def:DocumentRef, Description

Attribute	Usage	Allowable Values	Description
Type	Required	Value Description: Origin type • The list of allowable values is not extensible and is expected to be available as CDISC Controlled Terminology in the near future. Once available, the list of allowable values in the "Allowable Values" column will reference the C-Code and name of the "Origin Type" codelist. • See Section 4.3.2, Origin/Source/Traceability Considerations	Business Rule: If the variable is derived, a MethodDef must be provided.
Source	Conditional	Value Description: Data source • Source is required for SDTM and ADaM datasets when Origin Type is not Predecessor. Source is not required for SEND datasets. • See Section 4.3.2, Origin/Source/Traceability Considerations	Business Rule: The list of allowable values for each def:Origin/@Source value is not extensible and is expected to be available as CDISC Controlled Terminology in the near future. Once available, the list of allowable values in the "Allowable Values" column will reference the C-Code and name of the "Origin Source" codelist subsets.

5.3.13 CodeList Element

For each controlled terminology referenced by variable or valuelist, a CodeList element with the definition of the controlled terminology must be provided.

Element Name	CodeList
Element XPath(s)	/ODM/Study/MetaDataVersion/CodeList
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality:

	○ CodeList element must be provided for each distinct value of the CodelistOID attribute in a CodeListRef element in the MetaDataVersion. ○ If any ItemRef elements include a non-null value for the RoleCodelistOID attribute, there must be a CodeList element for each distinct value of the RodeCodelistOID in an itemDef element.
Attributes	OID, Name, DataType, def:IsNonStandard, def:StandardOID, SASFormatName, def:CommentOID
Child Elements	EnumeratedItem, CodeListItem, ExternalCodeList, Alias, Description

Attribute	**Usage**	**Allowable Values**	**Description**
OID	Required	Text	Unique ID for the CodeList. See Section 3.5.1, OIDs, for a discussion of OID values.
Name	Required	Text	• Name assigned to the CodeList element • Business Rule: The Name must be unique within the set of Codelist elements.
DataType	Required	"text", "float", "Integer"	The data type of the codes
def:IsNonStandard	Conditional Used when the controlled terminology includes a set of EnumeratedItem or CodeListItem elements as defined by the sponsor	"Yes"	Identifies the CodeList as sponsor-defined
def:StandardOID	Conditional Required in the context of a regulatory submission when the controlled terminology is drawn from a published CDISC standard	Text	• References OID of a def:Standard element with Type="CT" • Business Rule: Except for External CodeLists, each CodeList must have either a def:StandardOID attribute or a def:IsNonStandard attribute set to "Yes".
SASFormatName	Optional	Text	The SASFormatName Business Rules: • The SASFormatName must be a legal SAS format. • The SASFormatName needs to start with a "$" character when the CodeList DataType is "Text".
def:CommentOID	Optional	Text	• Reference to the unique ID of a def:CommentDef element that contains the comment for the CodeList element. See Section 5.3.15, def:CommentDef Element. • Business Rule: Must match the OID of a def:CommentDef element in the same MetaDataVersion.

5.3.13.1 EnumeratedItem Element

The EnumeratedItem element defines a CodedValue in a controlled terminology. Lists the CodedValues for all items in the controlled terminology.

Element Name	EnumeratedItem
Element XPath(s)	/ODM/Study/MetaDataVersion/CodeList/EnumeratedItem
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: Each CodeList element must contain either 1 or more EnumeratedItem elements, 1 or more CodeListItem elements, or 1 ExternalCodelist element. • Business Rules: ○ For Controlled Terminologies, where there is just a list of allowed values, an EnumeratedItem must be provided for each Item included in the Terminology. ○ The complete set of values relevant to the study must be provided regardless of whether they are referenced within the study data.
Attributes	CodedValue, Rank, OrderNumber, def:ExtendedValue
Child Elements	Alias, Description

Attribute	Usage	Allowable Values	Description
CodedValue	Required	Text	• The coded value • Business Rule: For NCI/CDISC Controlled Terminology, this must exactly match the CodedValue from the published Controlled Terminology ODM.
Rank	Optional	Integer	• Numeric significance of the EnumeratedItem relative to others in the CodeList • Business Rule: If this value is provided for any EnumeratedItem, it must be provided for all. **Note** that the Rank attribute does not imply a display order.
OrderNumber	Optional	Integer	• Display order of the item within the CodeList. • Business Rule: If this value is provided for any EnumeratedItem, it must be provided for all.
def:ExtendedValue	Conditional Required when the CodedValue is an extended value	"Yes"	Indicates a coded value that has been used by the sponsor to extend external controlled terminology **Note** that: • Controlled Terminologies should only be extended by the sponsor in case the Controlled Terminology allows extension, and only in the case where there is no equivalent value or synonym already in the CodeList. • The attribute should be omitted when the CodedValue is not an extended value because the only allowable value is "Yes".

5.3.13.2 CodeListItem Element

The CodeListItem element defines a CodedValue in a controlled terminology when a Decode value or Preferred Term is provided for each code. It lists the Coded Values and Decodes for all items in the controlled terminology.

Element Name	CodeListItem
Element XPath(s)	/ODM/Study/MetaDataVersion/CodeList/CodeListItem
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: Each CodeList element must contain either 1 or more EnumeratedItem elements, 1 or more CodeListItem elements, or 1 ExternalCodelist element. • Business Rules: ○ For Controlled Terminologies where there are Coded and Decoded values, a CodeListItem must be provided for each Item included in the Terminology. ○ The complete set of values relevant to the study must be provided regardless of whether they are referenced within the study data.
Attributes	CodedValue, OrderNumber, Rank, def:ExtendedValue
Child Elements	Decode, Alias, Description

Attribute	**Usage**	**Allowable Values**	**Description**
CodedValue	Required	Text	• The coded value • Business Rule: For NCI/CDISC Controlled Terminology, this must exactly match the CodedValue from the published Controlled Terminology ODM.
Rank	Optional	Integer	• Numeric significance of the CodeListItem relative to others in the CodeList • Business Rule: If this value is provided, it must be present for all CodeListItems with the same parent CodeList element. **Note** that the Rank attribute does not imply a display order.
OrderNumber	Optional	Integer	• Display order of the item within the CodeList • Business Rule: If this value is provided for any CodeListItem, it must be provided for all.
def:ExtendedValue	Conditional Required when the CodedValue is an extended value	"Yes"	Indicates a coded value that has been used by the sponsor to extend external controlled terminology **Note** that: • Controlled Terminologies should only be extended by the sponsor when the Controlled Terminology allows such extension, and only in the case where there is no equivalent value or synonym already in the CodeList. • The attribute should be omitted when the CodedValue is not an extended value, because the only allowable value is "Yes".

5.3.13.3 Decode Element

The Decode element defines a preferred term for a CodedValue in a CodeListItem.

Element Name	Decode
Element XPath	/ODM/Study/MetaDataVersion/CodeList/CodeListItem/Decode
Element Textual Value	None
Usage	• Requirement: Required • Cardinality: One • Other Information: This element is the container for a CodeListItem Decode value. The actual text value is provided in the child element TranslatedText.
Attributes	None
Child Elements	TranslatedText

5.3.13.4 ExternalCodeList Element

The ExternalCodeList element identifies the source of a third-party controlled terminology.

Element Name	ExternalCodeList
Element XPath	/ODM/Study/MetaDataVersion/CodeList/ExternalCodeList
Element Textual Value	None
Usage	• Requirement: Conditional. For controlled terminologies provided by third parties, an ExternalCodeList element must be provided to identify the name and version of the terminology. • Cardinality: One. Each CodeList element must contain either 1 or more EnumeratedItem elements, 1 or more CodeListItem elements, or 1 ExternalCodelist element. • Business Rule: Required for regulatory submissions to the FDA to provide the reference to the medical dictionaries used.
Attributes	Dictionary, Version, ref, href
Child Elements	None

Attribute	**Usage**	**Allowable Values**	**Description**
Dictionary	Required	Text CDISC Submission Value (=CodedValue) as listed in the NCI/CDISC Controlled Terminology, CodeList Dictionary Name, NCI Code C66788. This CodeList is extensible. Samples: COSTART, ICD, LOINC, MedDRA, SNOMED, WHOART, WHODD	The name of the external codelist
Version	Required	Text	The version designator of the external codelist
ref	Optional	Text	Reference to a local instance of the dictionary
href	Optional	Text	URL of an external instance of the dictionary

5.3.14 MethodDef Element

A MethodDef element provides details about a computational algorithm that is used as part of a variable or value definition.

Element Name	MethodDef
Element XPath(s)	/ODM/Study/MetaDataVersion/MethodDef
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: Required for each unique value of the MethodOID attribute within the MetaDataVersion • Business Rule: Must contain the child Description element and zero or more child def:DocumentRef elements • Other Information: ○ When used in a submission (ODM@def:Context="Submission"), the child Description or referenced document should reference variable names defined in the same Define-XML document. ○ Note that each distinct method is expected to have a unique MethodOID and can be referenced from different variables. ○ When the algorithm is provided in an external file the def:DocumentRef element provides a reference to the file and Description /TranslatedText element provides a short high level descriptive.
Attributes	OID, Name, Type
Child Elements	Description, def:DocumentRef, FormalExpression

Attribute	Usage	Allowable Values	Description
OID	Required	Text	Unique ID for the MethodDef. See ODM specification Section 2.11 for OID considerations.
Name	Required	Text	The Method name
Type	Required	Computation, Imputation	The Method type. Computation is intended to describe an algorithm to populate a value; Imputation is the process of replacing missing data with substitute values.

5.3.14.1 FormalExpression Element

The FormalExpression element enables providing machine readable code to compute or impute the value of an ItemDef.

Element Name	FormalExpression
Element XPath(s)	/ODM/Study/MetaDataVersion/MethodDef/FormalExpression
Element Textual Value	Text string
Usage	• Requirement: Optional • Cardinality: Zero or More • Business Rule: The FormalExpression must evaluate to the correct DataType of the ItemDef that is to be imputed or computed using the Method in the computer language specified under Context.

	- Other Information: The way that FormalExpression is to be combined with the rest of the code in the particular programming language specified in the Context attribute is outside of the scope of the Define-XML 2.1 specification.
Attributes	Context
Child Elements	None

Attribute	Usage	Allowable Values	Description
Context	Required	Text	A free-form qualifier to suggest an appropriate computer language to be used when evaluating the FormalExpression content

5.3.15 def:CommentDef Element

The def:CommentDef element provides short comments that are contained in the Define-XML document or long comments referenced in external documents. See Section 4.8, Comment Definitions. The def:CommentOID attribute in ItemDef must reference a valid def:CommentDef.

Element Name	def:CommentDef
Element XPath(s)	/ODM/Study/MetaDataVersion/def:CommentDef
Element Textual Value	None
Usage	• Requirement: Conditional • Cardinality: Required for each unique value of the def:CommentOID attribute within the MetaDataVersion • Business Rule: Must contain the child Description element and zero or more child def:DocumentRef elements • Other Information: ○ When the comment is provided in an external file, the def:leafID attribute of the def:CommentDef element must be included and the Description element can include a short descriptive reference to the external file. ○ **Note** that each distinct comment is expected to have a unique def:CommentOID and can be referenced from different variables.
Attributes	OID
Child Elements	Description, def:DocumentRef

Attribute	Usage	Allowable Values	Description
OID	Required	Text	Unique ID for the CommentDef. See ODM specification Section 2.11 for OID considerations.

5.3.16 def:leaf Element

The def:leaf element provides the information needed to locate an external document file referenced in the Define-XML.

Element Name	def:leaf
Element XPath(s)	/ODM/Study/MetaDataVersion/def:leaf /ODM/Study/MetaDataVersion/ItemGroupDef/def:leaf

Element Textual Value	None
Usage	• Requirement: Conditional (required in the context of a regulatory data submission) • Cardinality: One for each distinct document file referenced by a def:DocumentRef element and one for each dataset file referenced by a def:ArchiveLocationOID attribute in an itemGroupDef element.
Attributes	ID, xlink:href
Child Elements	def:title

Attribute	**Usage**	**Allowable Values**	**Description**
ID	Required	The leaf ID is based on the XML xs:ID datatype, which is a Non-Colonized Name; therefore, ID attributes must start with either a letter or underscore (_), and may contain only letters, digits, underscores, hyphens and periods.	• Unique ID for the def:leaf. See ODM specification Section 2.11 for OID considerations. • Business Rule: def:leaf ID attributes must be unique within the Define-XML document (i.e., there can be no 2 def:leaf elements with the same ID attribute).
xlink:href	Required	xsd:anyURI	• URL that can be used to identify the location of a document or dataset file relative to the folder containing the Define-XML file. If the file is not located in the Define-XML folder, a relative file path should be included. • Business Rule: For regulatory submissions to the FDA, the locations specified have to conform to locations allowed in the eCTD, in the Technical Conformance Guide for the relevant regulatory authority and the study data specifications. See Section 2.2, References.

5.3.16.1 def:title Element

Element Name	def:title
Element XPath(s)	• /ODM/Study/MetaDataVersion/def:leaf/def:title • /ODM/Study/MetaDataVersion/ItemGroupDef/def:leaf/def:title
Element Textual Value	Text with the label for the document or dataset. It provides for a document title that differs from the file name, as may be found with annotated CRFs or supplemental documents.
Usage	• Requirement: Required • Cardinality: One • Other Information: **Note** that the def:title element has no attributes or child elements.
Attributes	None
Child Elements	None

6 Global Element Ordering

Because following the order of elements is a part of conformance to the Define-XML, all of the elements are listed below in their correct order for the user's reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="define2-1.xsl"?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3"
     xmlns:def="http://www.cdisc.org/ns/def/v2.1"
     xmlns:xlink="http://www.w3.org/1999/xlink"
     xmlns:arm="http://www.cdisc.org/ns/arm/v1.0"
     FileOID="CDISC-Sample"
     ODMVersion="1.3.2"
     FileType="Snapshot"
     CreationDateTime="2018-11-15T11:01:00"
     def:Context="Submission">
    <Study>
        <GlobalVariables>
            <StudyName/>
            <StudyDescription/>
            <ProtocolName/>
        </GlobalVariables>
        <MetaDataVersion>
            <def:Standards>
                <def:Standard/>
            </def:Standards>
            <def:AnnotatedCRF>
                <def:DocumentRef/>
            </def:AnnotatedCRF>
            <def:SupplementalDoc>
                <def:DocumentRef/>
            </def:SupplementalDoc>
            <def:ValueListDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <ItemRef>
                    <def:WhereClauseRef/>
                </ItemRef>
            </def:ValueListDef>
            <def:WhereClauseDef>
                <RangeCheck>
                    <CheckValue/>
                </RangeCheck>
            </def:WhereClauseDef>
            <ItemGroupDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <ItemRef/>
                <Alias/>
                <def:Class>
                    <def:SubClass/>
                </def:Class>
                <def:leaf>
                    <def:title/>
                </def:leaf>
            </ItemGroupDef>
            <ItemDef>
```

```
                <Description>
                    <TranslatedText/>
                </Description>
                <CodeListRef/>
                <def:Origin>
                    <Description>
                        <TranslatedText/>
                    </Description>
                    <def:DocumentRef>
                        <def:PDFPageRef/>
                    </def:DocumentRef>
                </def:Origin>
                <def:ValueListRef/>
            </ItemDef>
            <CodeList>
                <Description>
                    <TranslatedText/>
                </Description>
                <EnumeratedItem>
                    <Alias/>
                    <Description>
                        <TranslatedText/>
                    </Description>
                </EnumeratedItem>
                <CodeListItem>
                    <Decode>
                        <TranslatedText/>
                        <Alias/>
                        <Description>
                            <TranslatedText/>
                        </Description>
                    </Decode>
                </CodeListItem>
                <ExternalCodeList>
                    <Alias/>
                </ExternalCodeList>
            </CodeList>
            <MethodDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <FormalExpression/>
                <def:DocumentRef>
                    <def:PDFPageRef/>
                </def:DocumentRef>
            </MethodDef>
            <def:CommentDef>
                <Description>
                    <TranslatedText/>
                </Description>
                <def:DocumentRef>
                    <def:PDFPageRef/>
                </def:DocumentRef>
            </def:CommentDef>
            <def:leaf>
                <def:title/>
            </def:leaf>
         <arm:AnalysisResultDisplays/>
        </MetaDataVersion>
    </Study>
</ODM>
```

7 Appendices

Appendix A: XML Schema

Define-XML Version 2.1 XML documents should reference (directly or indirectly) the following schema files:

Define-XML Schema	schema/cdisc-define-2.1 (top level folder)
	schema/cdisc-define-2.1/define2-1-0.xsd
	schema/cdisc-define-2.1/define-extension.xsd
	schema/cdisc-define-2.1/define-enumerations.xsd
	schema/cdisc-define-2.1/define-ns.xsd
ODM 1.3.2 Schema (including W3C schema files)	schema/cdisc-odm-1.3.2 (top level folder)
	schema/cdisc-odm-1.3.2/ODM1-3-2.xsd
	schema/cdisc-odm-1.3.2/ODM1-3-2-foundation.xsd
	schema/core/xlink.xsd
	schema/core/xml.xsd
	schema/core/xmldsig-core-schema.xsd

The Analysis Results Metadata Standard extension to Define-XML will work with Define-XML v2.1.

The following XML schema files define the Analysis Results Metadata Standard extension to Define-XML, and the changes that are needed to make the Analysis Results Metadata Standard extension to Define-XML work with Define-XML v2.1.

Analysis Results Metadata for Define-XML Schema	schema/cdisc-arm-1.0 (top folder)	
	schema/cdisc-arm-1.0/arm1-0-0.xsd	• Change the namespace from: xmlns:def="http://www.cdisc.org/ns/def/v2.0" to: xmlns:def="http://www.cdisc.org/ns/def/v2.1"
	schema/cdisc-arm-1.0/arm-extension.xsd	• Change the redefine from: <xs:redefine schemaLocation="../cdisc-define-2.0/define-extension.xsd"> to: <xs:redefine schemaLocation="../cdisc-define-2.1/define-extension.xsd">
	schema/cdisc-arm-1.0/arm-ns.xsd	• Change the namespace from: xmlns:def="http://www.cdisc.org/ns/def/v2.0" to: xmlns:def="http://www.cdisc.org/ns/def/v2.1" • Change the import from: <xs:import namespace = "http://www.cdisc.org/ns/def/v2.0" schemaLocation = "../cdisc-define-2.0/define-ns.xsd"/> to: <xs:import namespace =

		"http://www.cdisc.org/ns/def/v2.1" schemaLocation = "../cdisc-define-2.1/define-ns.xsd"/>

Appendix B: Visualizing Value-Level Metadata

B1: Viewing Value-Level Metadata as Value Lists

A value list (or valuelist) describes all of the possible definitions of the contents of a variable when different conditions hold true (e.g., different definitions of VSORRES when VSTESTCD = "TEMP" or when VSTESTCD = "HR").

As an example, in the Vital Signs domain the variables VSORRES, VSORRESU are defined as follows:

Name	Label	Type	Controlled Terms	Role
VSORRES	Result or Finding in Original Units	text		Result Qualifier
VSORRESU	Original Units	text	Units for Vital Signs Results (C66770)	Variable Qualifier

Because the Type and Controlled Terms for VSORRES and VSORRESU differ depending on the value of VSTESTCD, value-level metadata should be attached to both variables. In each case, the condition that determines the value definition is simply "Where VSTESTCD EQ '<value>'".

The value-level metadata for VSORRES shows that the Temperature variable was collected as a floating point and the Heart Rate was collected as an integer value:

Source Variable	Condition	Label	Type	Controlled Terms
VSORRES	VSTESTCD EQ "TEMP"	Temperature	float	
VSORRES	VSTESTCD EQ "HR"	Heart Rate	integer	

The value-level metadata for VSORRESU shows the allowable units when the Temperature and Heart Rate variables were collected:

Source Variable	Condition	Label	Type	Controlled Terms
VSORRESU	VSTESTCD EQ "TEMP"	Temperature	text	["C", "F"]
VSORRESU	VSTESTCD EQ "HR"	Heart Rate	text	["Beats Per Minute"]

Notes:

1. It is possible to describe more complicated conditions. For example, where VSTESTCD in "TEMP", "WEIGHT", "HEIGHT".
2. The tables in this section are intended to be illustrative and do not define requirements for Define-XML value-level metadata displays.

B2: Viewing Value-level Metadata as Slices

Slices are also defined by valuelists, but they are displayed in a way that illustrates the domain dataset point of view. Instead of showing all the possible values of a variable under different conditions (as a valuelist does), a slice shows what a dataset looks like under a specific condition. Using the preceding example, the Vital Signs domain could be shown as 2 slices: (1) where VSTESTCD="TEMP" (or a *TEMP slice*) and (2) where VSTESTCD="HR" (or an *HR slice*).

TEMP Slice: VS Domain Where VSTESTCD="TEMP"

Name	Label	Type	Controlled Terms
STUDYID	Study Identifier	text	
DOMAIN	Domain Abbreviation	text	['VS']
USUBJID	Unique Subject Identifier	text	
VSSEQ	Sequence Number	float	
VSTESTCD	Vital Signs Test Short Name	text	Vital Signs Test Code (C66741)
VSTEST	Vital Signs Test Name	text	Vital Signs Test Name (C67153)
VSORRES	Temperature	float	
VSORRESU	Temperature	text	['C']
VSSTRESC	Character Result/Finding in Std Format	text	
VSSTRESN	Numeric Result/Finding in Standard Units	float	
VSSTRESU	Standard Units	text	Units for Vital Signs Results (C66770)
VSBLFL	Baseline Flag	text	No Yes Response (C66742)
VISITNUM	Visit Number	float	
VSDTC	Date/Time of Measurements	datetime	ISO 8601 (Dates/Times)

HR Slice: VS Domain Where VSTESTCD EQ 'HR'

Name	Label	Type	Controlled Terms
STUDYID	Study Identifier	text	
DOMAIN	Domain Abbreviation	text	["VS"]
USUBJID	Unique Subject Identifier	text	
VSSEQ	Sequence Number	float	
VSTESTCD	Vital Signs Test Short Name	text	Vital Signs Test Code (C66741)
VSTEST	Vital Signs Test Name	text	Vital Signs Test Name (C67153)
VSORRES	Heart Rate	integer	
VSORRESU	Heart Rate	text	["Beats Per Minute"]
VSSTRESC	Character Result/Finding in Std Format	text	
VSSTRESN	Numeric Result/Finding in Standard Units	float	
VSSTRESU	Standard Units	text	Units for Vital Signs Results (C66770)
VSBLFL	Baseline Flag	text	No Yes Response (C66742)
VISITNUM	Visit Number	float	
VSDTC	Date/Time of Measurements	datetime	ISO 8601 (Dates/Times)

Most variables will share a common definition across all slices (e.g. --DOMAIN, --SEQ). Only variables which have a separate definition for each slice need be displayed separately for each slice. If a variable does not have a definition specific to a given slice (i.e., does not have a value definition for that slice), the definition of the parent variable is used. This minimizes the amount of metadata required to define slices.

Note: The tables in this section are intended to be illustrative; they do not define how slices should be shown.

Appendix C: CDISC Define-XML Team

This specification was developed by the CDISC Define-XML Team:

- Sam Hume, CDISC
- Sally Cassells, CDISC
- Lex Jansen, SAS Institute
- Marcelina Hungria, DIcore Group, LLC
- Kevin Burges, Formedix
- Dave Iberson-Hurst, Assero
- Jozef Aerts, FH Johanneum
- Monika Kawohl, HMS Analytical Software GmbH
- Jennifer Feldmann, Epreda
- Mike Molter, Wright Avenue Partners
- Valerie Williams, ICON

Appendix D: Deprecated Components

Some components of Define-XML v2.1 have been deprecated. A list of the deprecated elements and attributes is provided in the following table.

Element(s)	Component	Comment
MetaDataVersion	def:StandardName	The Standards and Standard elements replace the def:StandardName attribute used in Define-XML Version 2.0.
MetaDataVersion	def:StandardVersion	The Standards and Standard elements replace the def:StandardVersion attributes used in Define-XML Version 2.0.
ItemGroupDef	def:Class	The def:Class attribute in the ItemGroupDef element was replaced by a def:Class element. See Section 5.3.11.2, def:Class Element.