Macro Notes for

%DefineXML

in Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version** | **Date** | **Author** | **Description of change** |
| 1.0 | 30AUG2016 | Katja Glaß | Initial macro documentation notes |
|  |  |  |  |

Table of Contents

[Revision History 1](#_Toc460336069)

[1. Purpose of %DefineXML 3](#_Toc460336070)

[1.1 Rough process of DefineXML generation: 3](#_Toc460336071)

[2. Disclaimer 4](#_Toc460336072)

[3. Macro License (MIT) 4](#_Toc460336073)

[4. Macro Parameters 4](#_Toc460336074)

[5. Output Generation 5](#_Toc460336075)

[5.1 General structure of the define.xml output file 5](#_Toc460336076)

[5.2 Header information 6](#_Toc460336077)

[5.3 Documents information 7](#_Toc460336078)

[5.4 Datasets information 7](#_Toc460336079)

[5.5 Variables Information 9](#_Toc460336080)

[5.6 Value Level Metadata 11](#_Toc460336081)

[5.7 Codelist / Controlled Terms / Thesauri Information 15](#_Toc460336082)

[5.8 Computational Algorithms / Methods Information 17](#_Toc460336083)

[5.9 Comments Information 18](#_Toc460336084)

[6. References 19](#_Toc460336085)

[6.1 Define-XML-2-0\_ReleasePackage20140424 19](#_Toc460336086)

# Purpose of %DefineXML

According CDISC the define.xml is the following:

*“Define-XML transmits metadata for SDTM, SEND and ADaM datasets; it is the metadata file sent with every study in each submission, which tells the FDA what datasets, variables, controlled terms, and other specified metadata were used.”[[1]](#footnote-1)*

The CDISC DefineXML Version 2 is requested for study submissions from FDA and PMDA. To support this process the DefineXML macro creates the required define.xml file in the CDISC specified format out of define information which is used as input for the macro through multiple input datasets.

## Rough process of DefineXML generation:

The following graphic displays the rough process to get the DefineXML document. The red circle demonstrates the scope of this macro.



# Disclaimer

The views, opinions, interpretations, findings, and conclusions or recommendations expressed in the %defineXML documents are strictly those of the author(s). They do not necessarily reflect the views of any company. Examples and contents may or may not follow the standards like CDISC.

# Macro License (MIT)

Copyright (c) 2016 Katja Glaß

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Macro Parameters

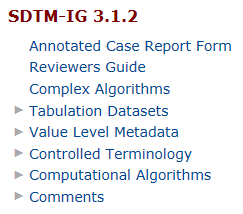
The options are provided to the macro through parameters.

| Parameter | Default | Description |
| --- | --- | --- |
| OUTFILE\* |  | Output file name where the created define.xml file is created (.xml extension is optional). |
| INDAT\_HEADER\* | header | SAS Input dataset containing the header information |
| INDAT\_DATASETS\* | datasets | SAS Input dataset containing the dataset information |
| INDAT\_VARIABLES\* | variables | SAS Input dataset containing the variable information |
| INDAT\_VALUELEVEL | valuelevel | SAS Input dataset containing the value level information |
| INDAT\_CODES | codes | SAS Input dataset containing the code list information |
| INDAT\_METHODS | methods | SAS Input dataset containing the method information |
| INDAT\_COMMENT | comments | SAS Input dataset containing the comment information |
| INDAT\_DOCUMENTS | documents | SAS Input dataset containing the documents information |
| \* required parameters | | |

# Output Generation

## General structure of the define.xml output file

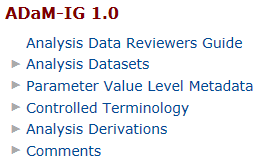
The define.xml output file is a pure text file in UTF8 format. This CDISC file comes together with a style sheet to be able to display the content in a browser. This document will only describe the display of the define.xml in combination with the style sheet. For detailed information about the text content of the define.xml, please refer to the official documentation referenced in 4.1 CDISC Define-XML Specification - Version 2.0. The screenshots use the CDISC examples from the define package (4.2 Define-XML-2-0\_ReleasePackage20140424).

The table of contents (left-top) in the define.xml contains two groups. The first group are the documents (no sub-lists), e.g. the Annotated Case Report Form which is required for SDTM and other documents like the Reviewers Guide, a Complex Algorithm documentation. Any documents can be included and referenced from the define.xml document. So also the Statistical Analysis Plan or any document can be included.

The second part contains the dataset and related descriptions which is splitted in multiple sections.

The “Tabulation Datasets” links to a dataset overview table. Below the single datasets are described.

The “Value Level Metadata” displays the value levels per dataset where these are specified for.

The “Controlled Terminology” contains next to CDISC controlled terminology also company specific terminologies and code lists. Thesauri are also described here.

The “Computational Algorithms” and “Comments” contains descriptions and comments which are also displayed at the linked areas, e.g. as dataset comments, variable derivation algorithms and similar.

The table of contents for ADAM looks slightly differently but contains the same content, just differently labeled, grouped and displayed in the whole XML document. These differences are only available in the display and controlled through the provided style sheet file.

## Header information

Depending on the used style sheet just a few or quite some header information are displayed in the XML web view. The header information are filled with some fixed values and some values taken from the header input dataset.

The header is created the following:

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

<?xml-stylesheet type="text/xsl"

StyleSheet

href="stylesheets/define2-0-0.xsl"?>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

OUTFILE (name part)

<!-- File: test\_001.xml -->

&sysuserid

<!-- Author: xxxxx -->

StudyDescription

<!-- Description: This is the define.xml for a study -->

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<ODM

SchemaLocation

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/v2.0"

ODMVersion="1.3.2"

FileOID="Study12345-defineXML-2.0"

FileOID

FileType="Snapshot"

Current datetime

CreationDateTime="2016-06-10T11:09:49"

Originator

Originator="Company XY"

SourceSystem="SAS Macro definexml"

Macro version

SourceSystemVersion="1.0">

StudyOID

<Study OID="Study12345">

<GlobalVariables>

StudyName

<StudyName>Study 12345</StudyName>

StudyDescription

<StudyDescription>This study has a description,

you might want to use here

</StudyDescription>

ProtocolName

<ProtocolName>ABC1234</ProtocolName>

</GlobalVariables>

„CDISC.“ + standard (whitout „\_IG) + „.“ + version

<MetaDataVersion OID="CDISC.SDTM.3.1.2"

Name="Study 12345, Data Definitions"

StudyName

Description="Study 12345, Data Definitions"

def:DefineVersion="2.0.0"

def:StandardName="SDTM-IG"

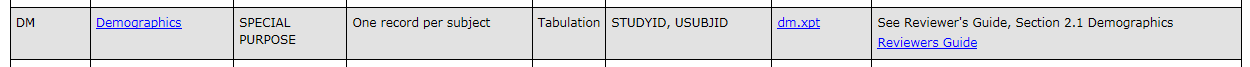
Standard

Version

def:StandardVersion="3.1.2">

## Documents information

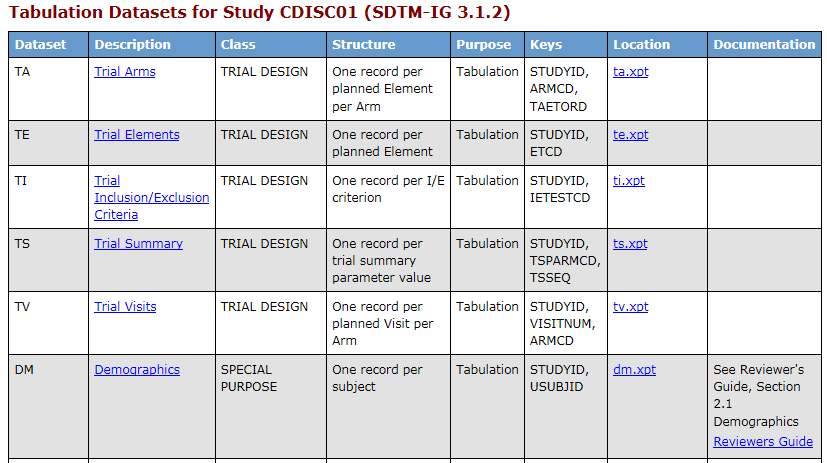
The definition of the documents appears in different places in the define.xml. First of all they appear in the table of contents (top-left). There the documents itself are linked. Furthermore the documents can be linked for any object, like datasets, variables, value level and similar which is done either through a comment or a method.



There are two options how a document can be linked. Either the document itself is used or a named destination within the document. To be able to use named destinations, this must be set up in the corresponding PDF document.

## Datasets information

The information about the datasets is displayed in the very first table. Some information is not displayed at all but is just available as object properties for a dataset. The “Keys” content is not coming from the DATASETS information, but from the VARIABLES information.



The following table shows which input information is used where:

| Variable Name | Variable Label | Usage in Display Table Column |
| --- | --- | --- |
| DOMAIN | Domain | Dataset |
| DESCRIPTION | Description | Description |
| CLASS | Class | Class |
| STRUCTURE | Structure | Structure |
| PURPOSE | Purpose | Purpose |
| ARCHIVELOCATION | Archive Location | Location |
| REPEATING | Repeating | Not displayed, just an XML attribute |
| ISREFERENCEDATA | Is Referenced Data | Not displayed, just an XML attribute |
| COMMENT | Comment OID | Documentation contains the description of the corresponding commentoid in the INDAT\_COMMENTS dataset |
| ORDID | Order for Domain Display | Not displayed, used to order the datasets for the display. Explanation: The dataset with the lowest order number is displayed as first item in the table. |

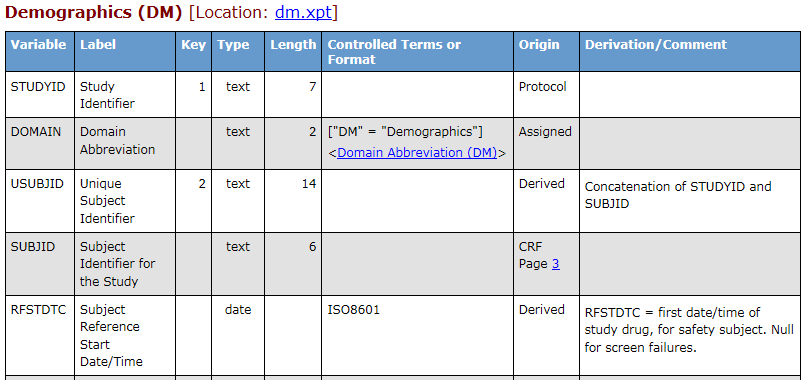
The folloing XML Attributes are set for the ItemGroupDef (dataset) definition:

|  |  |
| --- | --- |
| XML Attribute | Origin |
| OID | “IG.” + domain |
| [Domain | Domain] (only if standard is SDTM) |
| Name | Domain |
| SASDatasetName | Domain |
| Repeating | Repeating |
| IsReferenceData | IsReferenceData |
| Purpose | Purpose |
| Def:Strucutre | Structure |
| Def:Class | Class |
| Def:CommentOID | Comment |
| Def:ArchiveLocationID | “LF.” + last part of ArchiveLocation split by “/” |
| Description – TranslatedText | Description |
| ItemRef | See Table 1 – coming from VARIABLES information |
| Def:leaf - ID | “LF.” + last part of ArchiveLocation split by “/” |
| Def:leaf – xlink:href | ArchiveLocation + “.xpt” |
| Def:leaf – title | ArchiveLocation + “.xpt” |

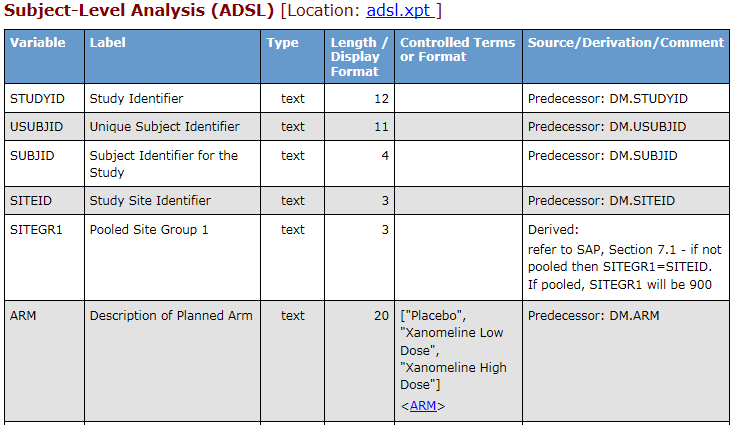
## Variables Information

The variable information is displayed per each dataset in a separate table. Some information is not displayed at all but is just available as object properties for a dataset. The “Keys” content is displayed in the dataset overview table. Depending on the “Purpose” of the dataset, the display table looks different.

Layout for Purpose = “Tabulation”:



Layout for Purpose = “Analysis”:



The following table shows which input information is used where:

| Variable Name | Variable Label | Usage in Display Table Column |
| --- | --- | --- |
| DOMAIN | Domain | Just as header of this table |
| VARNUM | Variable Number | Not displayed, used for variable ordering and as XML attribute |
| VARIABLE | Variable Name | Variable |
| LABEL | Label | Label |
| KEYSEQUENCE | Key Sequence | Key |
| TYPE | Type | Type |
| LENGTH | Length | Length |
| SIGNIFICANTDIGITS | Significant Digits | Not displayed, just an XML attribute |
| ORIGIN | Origin | Origin (for CRF additionally the CRF Page number is displayed) or  Source – depending on “Purpose” |
| CRFPAGE | CRF Page Number | Origin (for CRF Origin) |
| PREDECESSOR | Predecessor | Source after “Predecessor: “ for Origin = “Predecessor” |
| DISPLAYFORMAT | Display Format | Length / Display Format |
| COMPUTATIONMETHODOID | Computation Method Object ID | Derivation - contains the description of the corresponding ComputationMethod in the INDAT\_METHODS dataset |
| CODELISTNAME | Codelist Name | Controlled Terms or Format |
| MANDATORY | Manditory | Not displayed, just an XML attribute |
| VALUELISTOID | Value List Object ID | Used for linking “Variable” to this value list |
| COMMENT | Comment OID | Comment - contains the description of the corresponding commentoid in the INDAT\_COMMENTS dataset |

Table 1: XML Attributes used for ItemGroupDef definition

|  |  |
| --- | --- |
| XML Attribute | Origin |
| ItemRef – ItemOID | “IT.” + Domain + “.” + Variable |
| ItemRef – OrderNumber | varnum |
| ItemRef – Mandatory | mandatory |
| ItemRef – KeySequence | keysequence |
| ItemRef – MethodOID | computationmethodoid |

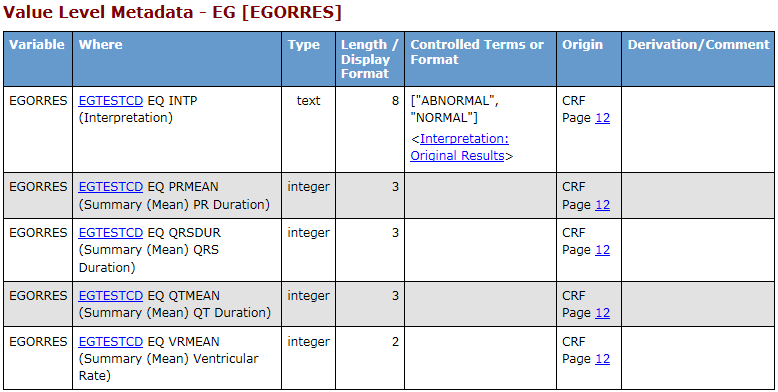
The folloing XML Attributes are set for the ItemDef (dataset-variables) definition:

| XML Attribute | Origin |
| --- | --- |
| ItemOID | “IT.” + Domain + “.” + Variable |
| Name | Variable |
| DataType | Type |
| [Length | Length] (if type in “'float' 'integer' 'text'”) |
| SASFieldName | Variable |
| SignificantDigitis | SignificantDigitis |
| def:DisplayFormat | displayformat |
| def:CommentOID | comment |
| Description – TranslatedText | Label |
| CodeListRef CodeListOID | “CL.” + codelistname |
| def:Origin Type | Origin |
| def:Origin def:DocumentRef leafID | (CRF origin) leafID of the CRF marked INDAT\_DOCUMENTS entry |
| def:Origin def:DocumentRef def:PDFPageRef PageRefs | (CRF Origin) CRFPage |
| def:Origin Description TranslatedText | (Predecessor origin) Predecessor |
| def:ValueListRef ValueListOID | ValueListOID |

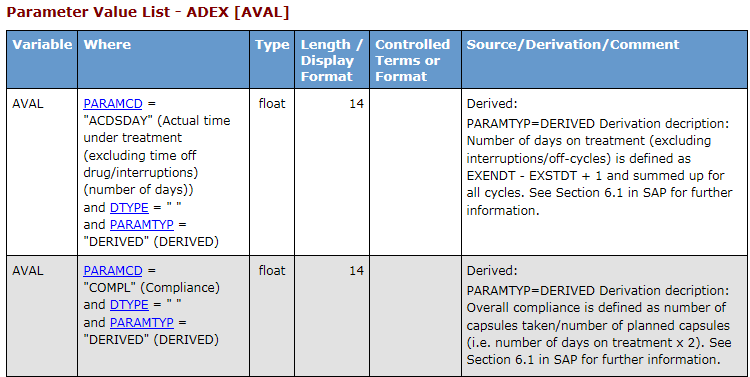
## Value Level Metadata

The value level metadata is displayed per variable within a separate table.

Layout for Purpose = “Tabulation”:



Layout for Purpose = “Analysis”:



The following table shows which input information is used where:

| Variable Name | Variable Label | Usage in Display Table Column |
| --- | --- | --- |
| VALUELISTOID | Value List Object ID | Not displayed, used as link ID |
| VALUEORDER | Value Order | Not displayed, used to order values |
| VALUENAME | Value Name | Variable |
| ITEMOID | Item Object ID | Not displayed, used for links |
| WHERECLAUSEOID | Where Clause Object ID | Where column is displayed by information coming from this column. A special format must be provided (see 3.4.5). |
| TYPE | Type | Type |
| LENGTH | Length | Length / Display Format |
| ORIGIN | Origin | Origin |
| CRFPAGE | CRF Page Number | Origin |
| PREDECESSOR | Predecessor | Source after “Predecessor: “ for Origin = “Predecessor” |
| COMPUTATIONMETHODOID | Computational Method | Derivation - contains the description of the corresponding ComputationMethod in the INDAT\_METHODS dataset |
| CODELISTNAME | Codelist Name | Controlled Terms or Format |
| SIGNIFICANTDIGITS | Significant Digits | Not displayed, just an XML attribute |
| DISPLAYFORMAT | Display Format | Length / Display Format |
| MANDATORY | Mandatory | Not displayed, just an XML attribute |
| DESCRIPTION | Description | Where – displayed in brackets as last information in this cell |
| COMMENT | Comment OID | Comment - contains the description of the corresponding commentoid in the INDAT\_COMMENTS dataset |

The value level metadata is stored in three different XML element types. Major information is stored in the ItemDef area. Additionally there is a related WhereClausDef required. And finally a corresponding ValueListDef is created.

The following XML Attributes are set for the ItemDef definition:

|  |  |
| --- | --- |
| XML Attribute | Origin |
| OID | Itemoid |
| Name | ValueName |
| DataType | Type |
| [Length | Length] (if type in “'float' 'integer' 'text'”) |
| SASFieldName | ValueName |
| SignificantDigitis | SignificantDigitis |
| def:DisplayFormat | DisplayFormat |
| def:CommentOID | Comment |
| Description - TranslatedText | Description |
| CodeListRef CodeListOID | “CL.” + CodelistName |
| def:Origin Type | Origin |
| def:Origin def:DocumentRef leafID | (CRF origin) leafID of the CRF marked INDAT\_DOCUMENTS entry |
| def:Origin def:DocumentRef def:PDFPageRef PageRefs | (CRF Origin) CRFPage |
| def:Origin Description TranslatedText | (Predecessor origin) Predecessor |

The following XML Attributes are set for the WhereClauseDef definition:

|  |  |
| --- | --- |
| XML Attribute | Origin |
| OID | Whereclauseoid |
| For all category and value pairs (part 3&4, 5&6, 7&8, …) | |
| RangeCheck SoftHard | "Soft" |
| RangeCheck Comparator | "EQ" |
| RangeCheck def:ItemOID | “IT.” + Part 2 + Part 3 / 5 / 7 / … |
| RangeCheck <CheckValue> | Part 4 / 6 / 8 / … |

**Missing:** If the <CheckValue> part is missing or contains the value “MISSING”, then the value is not filled for the <CheckValue> but is kept empty.

The following example demonstrates the fill-out. The OID is:

“IT.LB.LBTESTCD.GLUC.LBCAT.URINALYSIS.LBSPEC.URINE.LBMETHOD.DIPSTICK”

Then the RangeCheck would be:

<def:WhereClauseDef OID="WC.LB.LBTESTCD.GLUC.LBCAT.URINALYSIS.LBSPEC.URINE.LBMETHOD.DIPSTICK">

<RangeCheck SoftHard="Soft" def:ItemOID="IT.LB.LBTESTCD" Comparator="EQ">

<CheckValue>GLUC</CheckValue>

</RangeCheck>

<RangeCheck SoftHard="Soft" def:ItemOID="IT.LB.LBCAT" Comparator="EQ">

<CheckValue>URINALYSIS</CheckValue>

</RangeCheck>

<RangeCheck SoftHard="Soft" def:ItemOID="IT.LB.LBSPEC" Comparator="EQ">

<CheckValue>URINE</CheckValue>

</RangeCheck>

<RangeCheck SoftHard="Soft" def:ItemOID="IT.LB.LBMETHOD" Comparator="EQ">

<CheckValue>DIPSTICK</CheckValue>

</RangeCheck>

</def:WhereClauseDef>

The following XML Attributes are set for the ValueListDef definition:

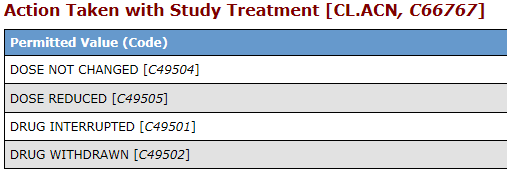
|  |  |
| --- | --- |
| XML Attribute | Origin |
| OID | ValueListOID |
| ItemRef ItemOID | ItemOID |
| ItemRef OrderNumber | ValueOrder |
| ItemRef Mandatory | Mandatory |
| ItemRef MethodOID | ComputationMethodOID |
| ItemRef def:WhereClauseRef WhereClauseOID | WhereClauseOID |

## Codelist / Controlled Terms / Thesauri Information

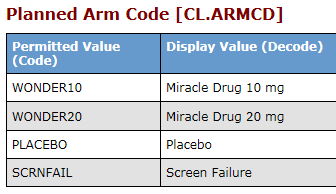
The Controlled Terms are displayed per list in separate tables and group external dictionaries in one table. The defineXML differentiates three types:

* Enumerated Item Lists
* Code/Decode Item Lists
* Dictionary Lists

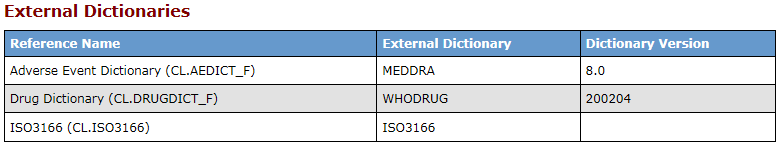
The following graphic displays an example of an “Enumerated” List:



The following graphic displays an example of a “Code/Decode” List:



The following graphic displays an example of “Dictionaries”:



The following table shows which input information is used where and for what:

| Variable Name | Variable Label | Usage in Display Table Column |
| --- | --- | --- |
| CODELISTNAME | Codelist Name | Used as ID with “CL.” as prefix |
| CODELISTLABEL | Codelist Label | Displayed everywhere the codelist is used (e.g. “Controlled Terms or Format” column in variable information table |
| RANK | Rank | Not displayed, just an XML attribute |
| CODE | External Code ID | “(Code)” column bracket value |
| CODEDVALUE | Coded Value | “(Code)” column |
| EXTENDEDVALUE\_NY | Extended Value of a CDISC Codelist? (N/Y) | “(Code)” column, bracket value of [\*] with the footnote “\* Extended Value” below the table if the value is set to Y |
| TRANSLATED | Translated Value | “(Decode)” column |
| TYPE | Type (DataType) | Not displayed, just an XML attribute |
| CODELISTDICTIONARY | Codelist Dictionary | “External Dictionary” |
| CODELISTVERSION | Codelist Version | “Dictionary Version” |
| DECODED | Decoded NY | Not displayed, just an XML attribute |
| OID\_POSTFIX | OID Postfix | Used as ID post fix, dot and value is combined and added to ID |

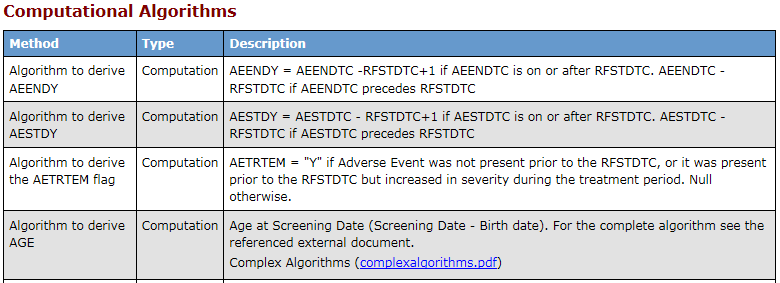
The folloing XML Attributes are set for the ValueListDef definition:

|  |  |
| --- | --- |
| XML Attribute | Origin |
| CodeList | “CL.” + CodelistName [+ “.” + OID\_POSTFIX] |
| Name | CodelistLabel |
| DataTyp | Type |
| EnumeratedItem: | (codelistdictionary = '' && DECODE ne “YES”) |
| CodedValue | CodedValue |
| OrderNumber | Rank |
| [def:ExtendedValue | “Yes”] (if UPCASE(extendedvalue\_ny) is in “Y”, “YES”) |
| Alias Name | Code |
| Alias Context | nci:ExtCodeID |
| CodeListItem: | (codelistdictionary = '' && DECODE eq “YES”) |
| CodedValue | CodedValue |
| OrderNumber | Rank |
| [def:ExtendedValue | “Yes”] (if UPCASE(extendedvalue\_ny) is in “Y”, “YES”) |
| TranslatedText | Translated |
| Alias Name | Code |
| Alias Context | nci:ExtCodeID |
| ExternalCodeList | (codelistdictionary ne '') |
| Dictionary | codelistdictionary |
| Version | codelistversion |

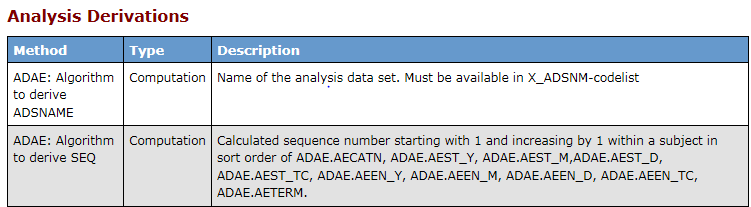
## Computational Algorithms / Methods Information

The computational algorithms are displayed all together in one table. The description could also contain links to PDF documents in general, in a specific contained section or links to a specific page.

Layout for Purpose = “Tabulation”:



Layout for Purpose = “Analysis”:



The following table shows which input information is used where and for what:

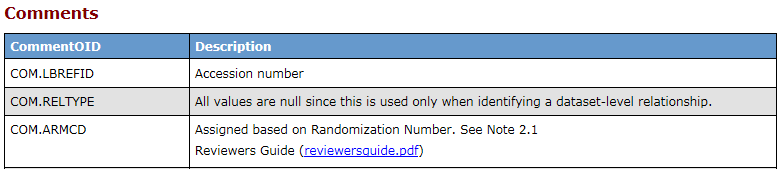
| Variable Name | Variable Label | Usage in Display Table Column |
| --- | --- | --- |
| COMPUTATIONMETHODOID | Computation Method Object ID | Not displayed, used as link ID |
| COMPUTATIONMETHODNAME | Computation Method Name | Method |
| COMPUTATIONMETHODTYPE | Computation Method Type | Type |
| DESCRIPTION | Description | Description |
| PDFLEAF | PDF Document Leaf ID | Included in Description displaying the description of the ID linked document, used as link |
| PDFPAGE | PDF Page | Links to specific page(s) in the PDF document |
| PDFDESTINATION | PDF Named Destination | Links to a specific named destination in the PDF document |

The folloing XML Attributes are set for the MethodDef definition:

|  |  |
| --- | --- |
| XML Attribute | Origin |
| OID | ComputationMethodoID |
| Name | ComputationMethodName |
| Type | ComputationMethodType |
| Description TranslatedText | Description |
| def:DocumentRef leafID | PDFLeaf (if provided) |
| def:PDFPageRef PageRefs | PDFPage (if provided) |
| Type | “PhysicalRef” |
| def:PDFPageRef PageRefs | PDFDestination (if provided and PDFPage not) |
| Type | “NamedDestination” |

## Comments Information

The comments are displayed all together in one table. The description could also contain links to PDF documents in general, in a specific contained section or links to a specific page.



The following table shows which input information is used where and for what:

| Variable Name | Variable Label | Usage in Display Table Column |
| --- | --- | --- |
| COMMENTOID | Comment Object ID | CommentOID |
| DESCRIPTION | Description | Description |
| PDFLEAF | PDF Document Leaf ID | Included in Description displaying the description of the ID linked document, used as link |
| PDFPAGE | PDF Page | Links to specific page(s) in the PDF document |
| PDFDESTINATION | PDF Named Destination | Links to a specific named destination in the PDF document |

The folloing XML Attributes are set for the CommentDef definition:

| XML Attribute | Origin |
| --- | --- |
| OID | ComputationMethodoID |
| Description TranslatedText | Description |
| def:DocumentRef leafID | PDFLeaf (if provided) |
| def:PDFPageRef PageRefs | PDFPage (if provided) |
| Type | “PhysicalRef” |
| def:PDFPageRef PageRefs | PDFDestination (if provided and PDFPage not) |
| Type | “NamedDestination” |

# References

## Define-XML-2-0\_ReleasePackage20140424

The following define package includes CDISC define examples, the style sheet and also the specification itself. The package can be accessed through the CDISC web page under the following link:

<http://www.cdisc.org/system/files/members/standard/define_xml_2_0_releasepackage20140424.zip>

1. CDISC – Define-XML definition (<http://www.cdisc.org/define-xml>) [↑](#footnote-ref-1)