# Define-XML 2.0 Stylesheet Documentation

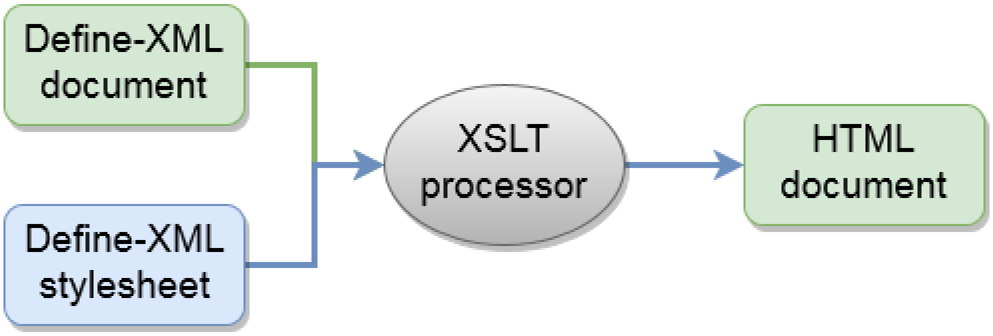
## 1. About

Define-XML is a standard developed by CDISC to support transmission of metadata for tabular datasets, such as SEND, SDTM, and ADaM datasets. This standard describes an XML format, which is used to create a Define-XML document containing all of the study metadata that has been modeled following a set of one or more CDISC model's IGs. One of the purposes of the Define-XML standard is to provide a specification for a machine-readable metadata and as a result it may be hard to read this document as a text. For this purpose, the XSLT language can be used to transform XML to a human readable format, like HTML. XSLT does not regulate the output format, but the stylesheet described in this document transforms XML to HTML which can be easily viewed in a browser.

In case of electronic submissions, regulatory agencies may require a stylesheet to be provided together with a Define-XML document. Please refer to regulatory agency guidelines for more details.[1]

## 2. Stylesheet usage

The transformation requires an XSLT processor which will execute the stylesheet code using Define-XML as a source and produce an HTML file. All modern web browsers have built-in processors for XSLT 1.0.



[1] E.g., FDA Study Data Technical Conformance Guide.

## 2.1. Parameters

For the Define-XML document development purposes, the stylesheet implements five parameters which can be set in an XSLT processor. See Appendix A for the full list of parameters available in the stylesheet and their description.

## 2.2. JavaScript

The stylesheet uses JavaScript to create interactive navigation bar which allows to expand and collapse menu and Value-Level Metadata (VLM) items, allowing to open external links in a different tab. The usage of JavaScript in the rendered file is kept to a minimum in order to make it compatible with old browsers.

# 3. Rendered Structure

The rendered version generated by the stylesheet is designed to be a human-readable representation of metadata stored in a Define-XML document and does not show all of the information kept in it or use the same structure as it is defined in the Define-XML standard. As an example, the OID attribute which is used to connect Define-XML elements is not generally shown in the rendered version.

Define-XML elements are ordered in accordance with the OrderNumber attribute where it is present. In other cases, the elements are ordered as they appear in the file.

## 3.1. Navigation Bar

The navigation bar is shown on the left side of the document and contains a list of sections generated in the rendered document. Some sections are expandable and contain links to individual tables within these sections. The expanding and collapsing is implemented using JavaScript and, in case the JavaScript functionality is disabled within a browser, all items in the navigation bar will be expanded.

The two buttons in the navigation bar allow to expand or collapse Value-level metadata in all datasets.

## 3.2. Header

The header contains information from the Study and MetadataVersion elements of the Define-XML document as well as the file creation date and version of the stylesheet.

As the **Metadata Description** attribute is optional, it is shown only in case it is present within the Define-XML document.

## 3.3. Analysis Result Metadata Summary (ADaM)

The Analysis Result Metadata Summary contains a list of analysis results grouped by analysis result displays.

## 3.4. Analysis Result Metadata Detail (ADaM)

The Analysis Result Metadata Detail table contains complete information for every analysis result described in the Define-XML document. The table is split by section, where each section corresponds for a specific analysis result.

## 3.5. Dataset Summary

The Datasets Summary section lists all datasets described in the Define-XML document and shows general information about them. The "*Keys*" column is generated from the list of variables which have a corresponding key attribute set (keySequence). There is no dataset property in the Define-XML 2.0 standard which contains a list of keys as they are shown in the rendered version.

Although not common, it is possible that one of the key variables is stored in a supplemental qualifiers dataset as a specific QNAM. In this case, the variable from the supplemental qualifiers dataset is shown in the "*Keys*" column of the parent dataset. For example, STUDYID, USUBJID, LBCAT, LBTESTCD, VISITNUM, QNAM.GROUP means that variable GROUP stored in the SUPPLB dataset (as QNAM = “”GROUP”) is a key variable.

## 3.6. Dataset Details

For each dataset, a separate table is created with the list of all variables and their details. For datasets with Value-level metadata (VLM) defined for at least one of its variables, both Variable- and Value-Level Metadata is shown in this table. When VLM is present in a dataset, an additional column named "*Where Condition*" is added to the table. Otherwise, the column is not shown. Each variable with VLM has the text 'VLM' in superscript next to its name on the right hand side. By clicking on the “VLM” text, the VLM is expanded or collapsed.

The "*Length or Display Format*" column shows the length attribute. When a variable has a display format attribute present, its value is shown instead of the length. In order to see length, display format and significant digits all at one, refer to the **displayLengthDFormatSD** parameter described in Appendix A.

Per CDISC standards the ISO8601 format should be used for variables having date, time, and duration related data types. As this is a format and not a codelist it is not stored in the Define-XML document and is added only by the stylesheet, based on the data type of the variable.

The "*Controlled Terms or ISO Format*" column contains the name of a codelist associated with the variable and a link to the corresponding section. By default, for codelists which have less than 6 terms, the full list of codes and decodes is shown in this column. The limit of the number of terms to be shown in this section is controlled by the **nCodeListitemDisplay** parameter.
The "*Origin / Source / Method / Comment*" column contains a combination of different attributes and elements as in many situations only some of them are populated and the main purpose of this column is to present a human-readable part of a variable definition . Note that Method can have a formal expression associated with it, which is only shown in the Methods section and not on this table.

As for many other objects in Define-XML, the where clause is stored in a normalized structure and not as a single condition. This means that each part of it, including conditions, variable IDs, comparator and values are stored separately. The stylesheet denormalizes this structure and combines all of these elements into a single line. In the Define-XML document, where clause variables are not referenced by their name, but by ItemDef OID. In general, variables from the where clause are included in the same dataset, and in this case a link to the variables is added to the final condition. But it is also possible that a variable is included in a different dataset, for example, when a where clause within the ADLB dataset refers to the variable COUNTRY from the ADSL dataset. In these situations, when a variable cannot be uniquely identified, the link is not added. The reason for that, is that in the Define-XML structure variable definitions are stored outside of the dataset definitions are can be referenced by more than one dataset.
In case the variable in a where clause has a codelist associated with it, decoded values from the codelist will be added to the condition values.

## 3.7. Supplemental Qualifiers Dataset Details

Supplemental qualifiers datasets store non-standard SDTM variables in a vertical structure. This vertical structure is expected to be described using a value-level metadata for variable QVAL where each value-level is defined by a simple condition "WHERE QNAM = XXX". Because of that, a layout of the supplemental dataset details table is slightly different from the layout of other dataset detail tables and shows metadata for those transposed vertical variables similar to other variables. There is no separate Where Clause column in this table as the "*Name*" column contains names for both standard SUPP variables (such as QVAL and IDVAR) and transposed vertical variables.

In some situations it is possible that variables which were transposed and are stored in the SUPP dataset have value level metadata defined for them back in the main dataset. In this case the "*Name*" column will also contain the where clause of these value-levels.

## 3.8. Controlled Terminology

The Controlled Terminology section contains codelists specified in the Define-XML document and description of external codelists or dictionaries referenced within the document. In case a codelist has decodes associated with the codelist terms, the decode will be shown in the second column. If codelist values have a rank assigned to them, the rank value will be shown in the third column. When the codelist terms are linked to NCI codes using Alias element, then the identificator values will be shown in brackets next to the codelist terms and codelist name.

## 3.9. Methods

The Methods section contains a list of all methods used within the Define-XML document. Methods can be reused within the document and referenced by more than one variable, but will appear only once in this table. The table lists method name, type, and description. If there is a code associated with a method, stored in the FormalExpression element, it will be shown in the description column.

## 3.10. Comments

The Comments section is not shown by default and is added only when the displayComments parameter is enabled. This section lists of all comments specified within the Define-XML document. As comments do not have a name attribute, the OID associated with a comment is listed in this table. Note that OIDs have a technical purpose within the Define-XML document and there is no naming convention for the OID values.

# 4. Rendering HTML

To see a rendered version it is sufficient to open a Define-XML file in a browser with a properly referenced stylesheet. It is also possible to render an HTML file using XSLT processors. There are multiple XSLT processors available on the market, both commercial and free. A list of XSLT processors is available on Wikipedia[2]. As an example, for Windows a Saxon XSLT processor can be used to render an HTML file from XML using the stylesheet. For macOS and Linux a free libxslt library is available which can perform this task.
When using XSLT processors it is possible to specify values for parameters listed in Appendix A.

[2] https://en.wikipedia.org/wiki/XSLT#Processor_implementations

# 5. Printing

The stylesheet implements a special style for printing purposes. It is recommended to select landscape page orientation when printing the document. The list of sections in the printed version, as well as each section's content is kept as much as possible the same as in the HTML version viewed in a browser. One noticeable difference is that all links shown together with the actual link paths. For example, a link to a reviewer's guide page 10, shown as following:
Reviewer's Guide [ 10 ]
Is shown as
Reviewer's Guide [ 10 ./path/to/file/adrg.pdf#page=10]

It is sometimes expected that the stylesheet printing functionality can be used to create a PDF file with interactive Table Of Contents, bookmarks, clickable links and other features which were expected in a submission of Define.pdf before Define-XML version 2.0. None of this is a purpose of the printing functionality. Discuss with a regulatory agency if define.pdf is needed for the submission. For example, FDA does not require define.pdf to be submitted in case the Define-XML document can be printed[3]:
*In addition to the define.xml, a printable define.pdf should be provided if the define.xml cannot be printed. To confirm that a define.xml is printable within the CDER IT environment, it is recommended that the sponsor submit a test version to cder-edata@fda.hhs.gov prior to application submission.*

# 6. Common Issues

### When opening a Define-XML document, the contents look as an unformatted text.

This means that the stylesheet was not loaded. Verify that the stylesheet file is located in the correct folder and that HREF attribute of the <xml-stylesheet> element (usually located on the second line) points to the correct stylesheet.

### Navigation bar menus do not work

Usually this happens when JavaScript is disabled in a browser. JavaScript is required for some of the stylesheet functionality, that is why it should be enabled when using the stylesheet.

### PDF links with a named destination do not work

Can be caused by whitespace characters in the named destination. If this is the case, whitespace characters must be properly escaped[4].
Example: NOTE#20420 will result in the PDF viewer as NOTE 420

[3] FDA SDTCG Section 4.1.4.5 Data Definition Files for SDTM, SEND, and ADaM.
[4] See PDF 1.7 specification Section 7.3.5 Name Objects for more details.

# 7. Contact Information

If you have any questions or suggestion, please feel free to write to define.xml.stylesheet@gmail.com

# 8. Development Team

| Name | Company |
|---|---|
| Lex Jansen | SAS Institute |
| Marcelina Hungria | DIcore Group |
| Rob Wartenhorst | GSK |
| Nicola Tambascia | PRA Health Sciences |
| Herve Guimard | Sanofi |
| Dmitry Kolosov | PAREXEL |

# 9. References

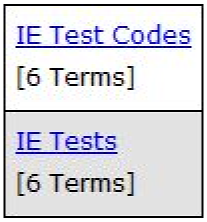
- W3C XML Transformation
- CDISC Define-XML Standard

# 10. Appendices

## Appendix A - Stylesheet parameters

### nCodeListItemDisplay

The number of codelist items which is displayed in the "*Controlled Terms or Format*" column of the dataset view is controlled by the nCodeListItemDisplay parameter. By default it is set to 5 and only codelists with 5 or less codes are



| |
|---|
| IE Test Codes<br>[6 Terms] |
| IE Tests<br>[6 Terms] |

shown with values in this column. In case a codelist has more than nCodeListItemDisplay items, it will be shown as
[XX Terms] in that column.

## displayMethodsTable (0/1)

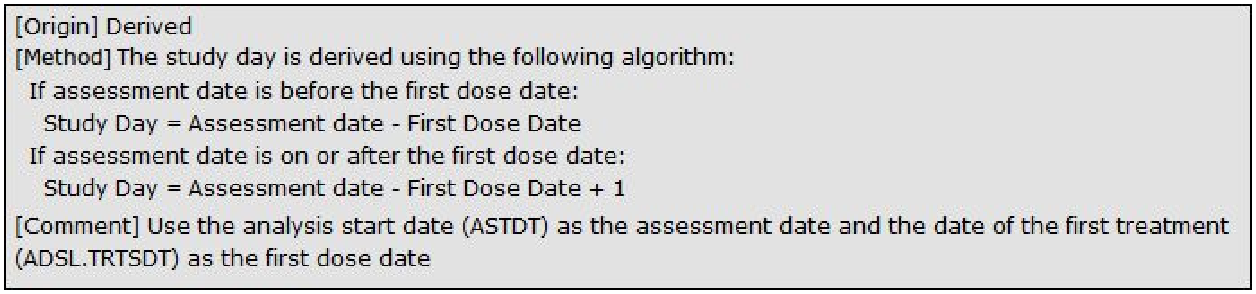
By default all algorithms are shown in a separate Methods table, which includes all of the methods described in the Define-XML document. This parameter allows to control whether this table is displayed or not.

## displayCommentsTable (0/1)

By default comments are not shown in a separate Comments table, which includes all comments described in the Define-XML document. This parameter allows to control whether this table is displayed or not.

## displayPrefix (0/1)

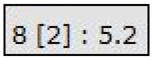
The "*Origin/Source/Method/Comment*" column combines different Define-XML attributes and elements. By setting this parameter to 1, prefixes [Comment], [Method], [Origin] will be displayed alongside with the descriptions, allowing to identify where it came from.



```
[Origin] Derived
[Method] The study day is derived using the following algorithm:
  If assessment date is before the first dose date:
    Study Day = Assessment date - First Dose Date
  If assessment date is on or after the first dose date:
    Study Day = Assessment date - First Dose Date + 1
[Comment] Use the analysis start date (ASTDT) as the assessment date and the date of the first treatment
(ADSL.TRTSDT) as the first dose date
```

## displayLengthDFormatSD (0/1)

When there is a format attribute provided, it will be shown in the "*Length or Display Forma*t" column, otherwise only the length will be shown in this column. When this parameter is set to 1, name of the column will be changed to "*Length [SignificantDigits] : Display Format*" and all of the corresponding attributes, when present in the Define-XML document, will be shown in this column.



8 [2] : 5.2