SEND Implementation Wiki - Define Fundamentals

What is the Define File?

The "define" file is a file which describes information about the SEND datasets, such as which domains are represented, which fields are present in each domain, usage of CT, and special calculations or notes on the population or calculation of fields.

The define file has two primary benefits:

- Allows for human-readable information on the contents of a SEND package
- Allows electronic systems consuming SEND packages to get an electronic "explanation" of the datasets' location and contents.

As of define 2.0, define is submitted as an xml file (previous versions allowed PDF).

Where Do I Find Specification?

- http://www.cdisc.org/define-xml - Site containing resources for the define-XML standard, such as the define spec, example files, xsl document, etc.

Structural Basics

Key Concepts

The majority of the content of the define.xml file consists of the specifications for domains and variables.

Variables are defined through **ItemDef** elements, usually 1 per variable per domain, although shared columns like STUDYID and USUBJID require only 1 definition regardless of how many domains use them.

Domains are specified as **ItemGroupDef** elements, which are in turn collections of ItemRef elements, or references to the variables' ItemDef elements.

Another key aspect to the define.xml includes CT, included as CodeList elements. These are referenced by the ItemDef (variable) elements through a CodeListRef subelement. CodeLists can either be a single reference to an external code list (e.g., SEND CT) or an itemized list of terms for a sponsor-specific list.

Element Overview

The following is a summary of the primary elements contained within a define.xml file.

First, header/framing elements (each consecutive element is a child of the one before it):

- **ODM** element, containing the datetime of file creation (also, some static references)
- **Study** element, which will frame the rest of the content, including a GlobalVariables element which provides the study ID, title, etc.
- **MetaDataVersion** element, which describes the standards used, etc.

Next, under the header elements, the following elements are used to describe the majority of the define content:

- **ValueListDef** elements, 1 for each custom list of values, such as enumerating the columns in a SUPP-- file
  - **ItemRef** subelements, 1 for each item attributed to the ValueListDef, in turn referencing an ItemDef (defined later)
- **ItemGroupDef** elements, 1 for each domain
  - **ItemRef** subelements, 1 for each variable in the domain and specifying internal-to-the-domain attributes, such as the ordering in the domain and so on. These elements in turn reference an ItemDef (defined later)
  - def:leaf element, 1 for the domain, describing the file to which the domain is associated.
- **ItemDef** elements, 1 for each variable used in any of the domains. Common columns, such as STUDYID, can be defined once and referenced within each domain, as they are used identically across domains, but outside of these cases, there is typically 1 ItemDef per column per domain. The ItemDef element's attributes describe the variable, including type, name, length, comments, and so on.
  - **CodeListRef** subelement, 0 or 1, describing the codelist (e.g., CT list) to which the ItemDef adheres (if applicable). This element in turn references the corresponding CodeList element (defined later)
- **CodeList** elements, 1 for each codelist used across any of the variables.
  - **ExternalCodeList** subelement, 0 or 1, used to reference the corresponding SEND CT list (most common case)
  - **CodeListItem** subelements, 0 to many, used for sponsor-specific lists of terms. These elements in turn have subelements for decodes and so on; reference the define specifications for details.

Please see the [1] spec for details on any of the elements noted.

Viewing

Opening a define.xml plain (e.g., with notepad) will just show raw xml, which is readable to robots.

If you want to view a define file in a more human-friendly way, you can use a style sheet, which is like a companion file that gives instructions to a browser on how to represent the xml file in a nice way.

Style sheets can take on many flavours, and no one style sheet out there is "definitive" - it is mainly a matter of preference.

From time to time, the CDISC XML Technology team publishes a define.xml style sheet for public use.
They can be found here: https://wiki.cdisc.org/display/PUB/Stylesheet+Library

Preparation

Getting a Base File

If you are using a vendor solution to create SEND files, it typically will come bundled with functionality to output a define.xml file. There are also third party standalone define.xml products.

If you need to create one yourself, then you have options:

- The Visual Define-XML Editor tool provided through the CDISC Open Source Alliance provides an easy way to create/edit your define.xml
- The Pinnacle21 Community tool can also be used to generate the basis of a define.xml file off a set of SEND XPT files (select "Generate Define.xml" and then your SEND XPT files).

These are good as a good starting point for your define.xml, as it will create all of the structural basics for you; however, it does not have the ability to populate the company-specific information such as comments, desired data types, custom controlled terminology, and so on.

Another option is to use an example define file provided from the define-xml site. These have more realistic examples, although not SEND-based.

Refining

The raw define file needs several additions and refinements.

Every study

- File/study metadata:
  - Creation datetime in the ODM element's CreationDateTime attribute
  - Study Name, Description, and Protocol Name in the Study element's GlobalVariables subelement (e.g., ABC123, 28-Day Oral Toxicology Study in Rats, and ABC123, respectively) - see the Technical Conformance Guide for additional direction on what the FDA expects, if this package is for a submission
  - Name under the MetaDataVersion element's Name attribute (e.g., "Study ABC123, Data Definitions")
- Domain (ItemGroupDef) keys under the def:DomainKeys attribute
- Domain variable references' (ItemRef) attributes, including a review of the Mandatory and Role attributes for domain variables (can be incorrect in the templates)
- Variable definitions' (ItemDef) attributes, including:
  - Populating the Origin attribute
  - Populating the Comments attribute
  - Revising the Type and/or Length attributes
  - Added a CodeListRef subelement when CT applies to the variable
- CodeList elements for each CT list used (internal and external)
- Any variable with an origin type=Derived must have a documented derivation method

Case by case:

- Any value lists used (ValueList), including SUPP-- variable descriptions.
- Any decodes for variables where a coded value needs a decode to make sense of the value should be specified via CodeListItems
- Any study-specific comments added as desired (SENDIG occasionally calls out some cases where comments are useful)

Style sheet

If you want the recipient to be able to view the define in the same presentation as you, then make sure to include the style sheet you use in the package. This is not required, but can be helpful.

Advance Define Concepts

Value-level Metadata

Value-level metadata needs to be defined when data in all rows of a variable cannot be described by a single collection of metadata.

Using the LB domain as an example, the LBORRES variable contains both qualitative and quantitative test results. The quantitative results may be integers or floating point values, and the floating point values may have different precisions. Some data may be collected and some derived. The qualitative results may use different result coding schemes that need documenting in different codelists. Thus, LBORRES cannot be described with a single collection of metadata at the variable level, and value-level metadata is required

All of the attributes and child elements (data type, length, significantdigits, codelist, origin, derivation method, comments) available for variable-level metadata are also available for value-level metadata. Additionally, value-level metadata needs to have some qualifier that describes the subset of data that is being described. Continuing the example of LBORRES, using the entry in LBTEST or LBTESTCD might be a good way to break up the values in the dataset into subsets which allow LBORRES to be defined with one set of metadata per test. In other words, the values in LBORRES could be described separately for each test (LBORRES values for RETI could be described separately from LBORRES values for GLUC...).

As of define-XML 2.0, multiple variables can be used, with different comparator operators, to create a WhereClause that identifies subsets of the dataset, e.g., creating a WhereClause that allows you to define metadata for LBORRES when LBTEST=PROT AND LBSPEC=URINE.
(Note: previously, in define-XML 1.0 it was only possible to identify one variable use to break up dataset values into subsets).

Extending CT

When official SEND CT is extended (values exist on study beyond those in the codelist), you'll need to provide these extensions in the define.

Best practice is to list only the items relevant to the study (not many CDISC codelists are applicable in their entirety to a study). Use the "Alias Name" element for published terms (using the term's C code as the value) and the "ExtendedValue" flag set to "Yes" for extended terms used on study.