PharmaSUG 2021 - Paper EP-143

Customizing define.xml files

Steffen Müller, mainanalytics GmbH

1 ABSTRACT

A metadata description following the Define-XML standard is a key component of the electronic data submission package that is sent to the health authorities in drug approval processes.

Since those authorities, e.g., FDA and PMDA, have differing requirements, it is useful to have a process in place that is able to convert a submission package from one standard to the other in a mostly automated way. An update of the contained controlled terminology or other components could also be a reason to update an existing submission package.

This paper shows how such a conversion could be done using standard tools such as SAS, Excel, and Pinnacle 21 Community, focusing on the update of the contained define.xml file.

2 INTRODUCTION

For electronic data submissions, health authorities request a package with:

- SAS datasets in the version 5 XPT format
- A define.xml file containing the submission metadata
- A stylesheet to display the define.xml content in a web browser
- Additional documentation, e.g., an annotated CRF and a Reviewer's Guide providing complementary information to the define.xml content

The define.xml file is an important part of the submission package and describes methods, formats, terminologies, and dictionaries used for the analysis.

Changes in the submission datasets usually also require an update of the define.xml. It is not sufficient to just replace the dataset but an adjustment of the define.xml metadata is necessary to create a new submission package.

The reason for an update might be, for example, that a health authority requests different laboratory units in a dedicated new dataset or an update due to using a new Implementation Guide for the Trial Summary domain is required.

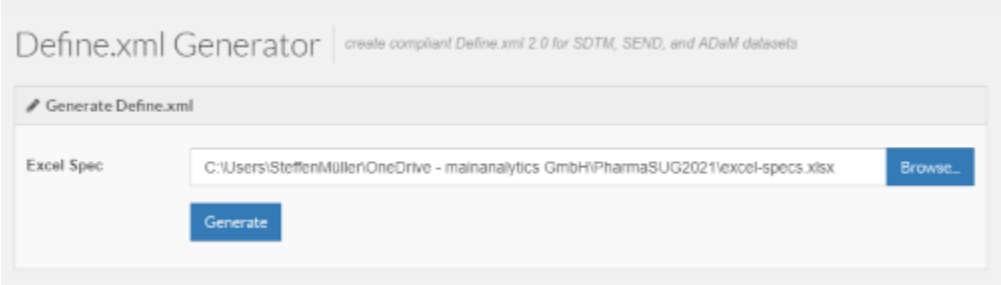
If such updates are necessary for various studies with a similar design, it is convenient to have a semi-automatic process in place.

The following sections describe how such an update of both, datasets and metadata, can be done in a consistent way.

3 PINNACLE 21 COMMUNITY TOOL FUNCTIONALITY

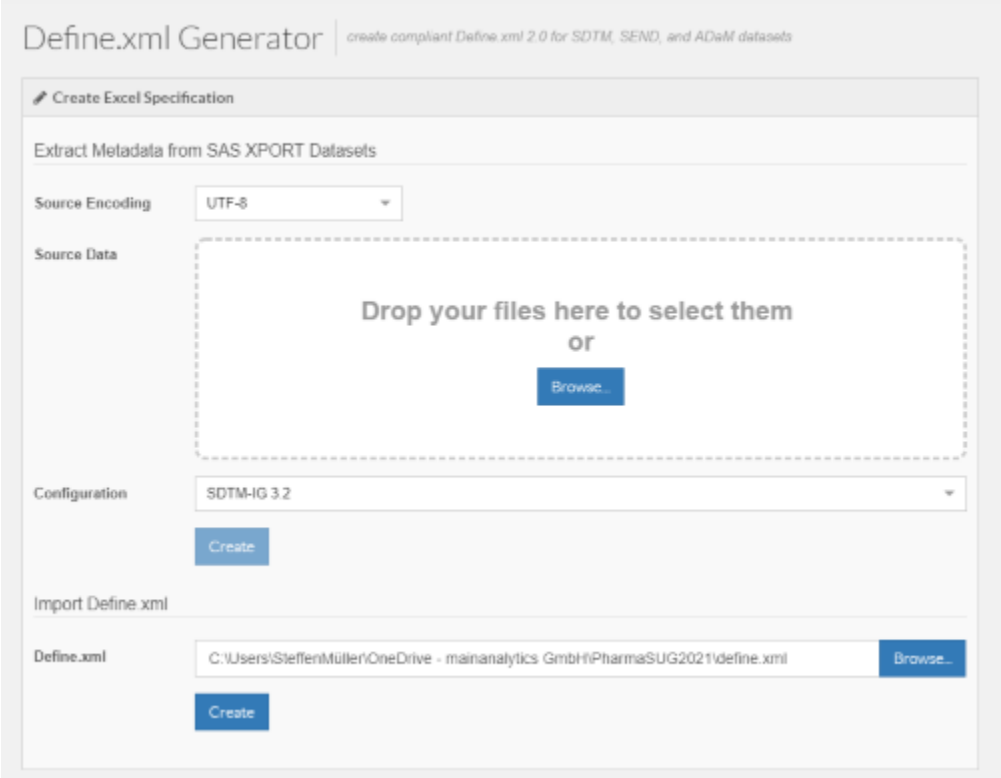
The following functionalities are used through the update process:

- The define.xml file can be generated with the Pinnacle 21 Community tool based on specifications in Excel format as described in “EXCEL SPECIFICATION STRUCTURE”.



Display 1 Pinnacle 21 Community “Generate define.xml” function

- The Pinnacle 21 Community tool also provides the functionality to convert a define.xml content into Excel specifications.
- A third functionality used in the update process creates Excel metadata specifications from SAS transport files. These specifications only contain dataset and variable names and labels. Further metadata like terminologies or value level information cannot be extracted to the Excel specifications since they are not contained in the SAS XPT files.



Display 2 Pinnacle 21 Community “Generate specification” function

3.1 EXCEL SPECIFICATION STRUCTURE

Excel specifications generated by and also used as input for the Pinnacle 21 Community tool contain the following sheets:

- **Study**

  General study information and used standards

- **Datasets**

  Contains all datasets, their labels and key variables

- **Variables**

  Contains all variables with type, length, label, format, controlled terminology

- **ValueLevel**

  Describes the value of a variable dependent on a where condition (“WhereClauses” sheet)

- **WhereClauses**

  The conditions which are used to describe a variable value (“ValueLevel” sheet)

- **Codelists**

  Codelists are associated with variables to describe all possible variable values. Some codelists also contain decodes for a better interpretation

- **Dictionaries**

  Contains names and versions of the used dictionaries

- **Methods**

  Contains variable derivations.

- **Comments**

  Contains variable comments.

- **Documents**

  References to documents contained in the submission package (e.g. annotated CRF or Reviewer’s Guide)

Some sheets contain an “ID” column so that entries can be linked over the sheets (e.g., a codelist can be linked to a variable).

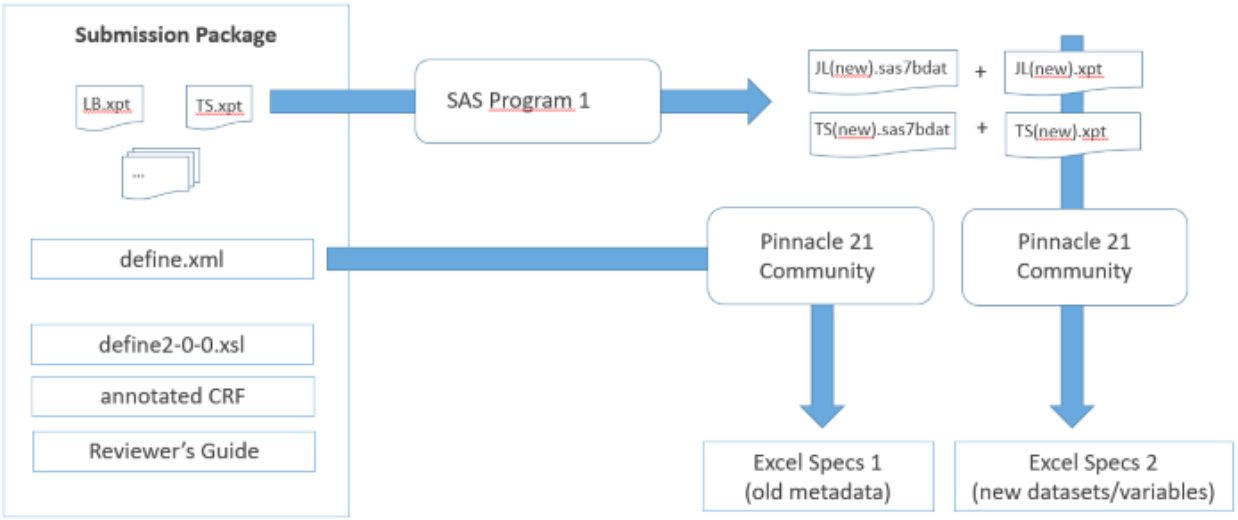
Order	Dataset	Variable	Label	Data Typ	Lengt	Significant Digits	Format	Mandator	Codelist	Origin	Pages	Method
1	LB	STUDYID	Study Identifier	text	7			Yes		Protocol		
2	LB	DOMAIN	Domain Abbreviation	text	2			Yes	LB.DOMAIN	Assigned		
3	LB	USUBJID	Unique Subject Identifier	text	14			Yes		Derived		USUBJID
4	LB	LBSEQ	Sequence Number	integer	2			Yes		Derived		SEQ
5	LB	LBREFID	Specimen ID	text	7			No		eDT		
6	LB	LBTESTCD	Lab Test or Examination S	text	7			Yes	LBTESTCD	Assigned		
7	LB	LBTEST	Lab Test or Examination N	text	22			Yes	LBTEST	eDT		
8	LB	LBCAT	Category for Lab Test	text	10			No		eDT		
9	LB	LBORRES	Result or Finding in Origin	text	8			No		eDT		
10	LB	LBORRESU	Original Units	text	7			No	LBRESU	eDT		
11	LB	LBORNRLO	Reference Range Lower Li	text	8			No		eDT		

Display 3 Sheet “Variables” for LB domain

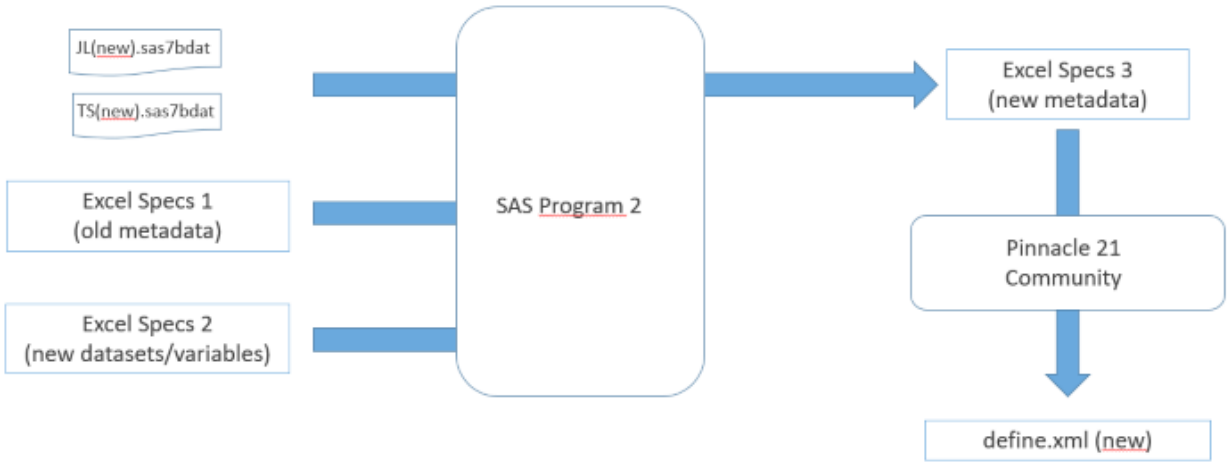
4 SUBMISSION PACKAGE UPDATE WORKFLOW

1. Convert the SAS transport files to SAS datasets. Update the datasets as necessary and/or create new datasets with a SAS program.

   Convert the SAS datasets back to SAS transport files after finalization of the update process.
2. Create Excel specifications with Pinnacle 21 Community based on the existing define.xml.
3. Create further Excel specifications based on the updated/new SAS datasets. These specifications only contain the dataset metadata. Further information like controlled terminologies or value level metadata are not available.
4. Update Excel specifications based on the existing define.xml.
   a. Integration of the specifications based on the new/updated datasets
   b. Content integration of the new/updated datasets, e.g., where clauses and value level metadata
5. Create a new define.xml based on the updated Excel specifications.
6. Create a new submission package containing the new/updated and unchanged components.



Display 4 Submission package update steps 1 to 3



Display 5 Submission package update steps 4 to 6

5 EXCEL SPECIFICATION UPDATE

The specification update is the most complex part of the process and includes creation of metadata based on the new/updated domains and merging them into the already existing metadata.

A SAS program was developed to implement these updates.

Note: If “worksheet” and “columns” are mentioned in the following sections then the respective SAS datasets and variables are meant. The program is working with the imported information and not directly accessing the Excel sheets.

5.1 SPECIFICATION IMPORT

The Excel specifications are imported via PROC IMPORT into SAS.

5.2 METADATA WORKSHEET UPDATE

The information contained in the sheets “Study”, “Documents” and “Dictionaries” usually remain unchanged but must be read in to be included in the final Excel specifications.

5.2.1 Worksheet “Datasets”

An update is only necessary if new datasets are included or datasets are deleted. For the first case the new dataset’s metadata is taken from the Excel specifications based on the new/updated datasets.

5.2.2 Worksheet “Variables”

For new variables, additional records will be created in the “Variables” worksheet. The following metadata columns will be taken from the Excel specifications based on the new/updated datasets:

- Dataset
- Variable
- Label

- Datatype
- Length
- Significant Digits
- Mandatory
- Role

For the other columns (Format, Codelist, Origin, Pages, Method, and Comment), no data exist in the Excel specifications.

There are various ways to fill these columns: If a new dataset must be created that is very similar to an existing one, then the existing metadata can be used as base.

Another possibility would be to create this information in the program, e.g.

```
if variable = "JLSTNRLO" then do;
    origin = "Derived";
    method = "JLSTNRLO";
 end;
```

or to read in another Excel file.

All these possibilities are more tedious than updating the specifications manually but are essential if an automatic process should be used.

5.2.3 Worksheet "WhereClauses"

Where clauses are used to describe the conditions under which the definition of a value applies.

For new/updated datasets the where clause information can be taken directly from the data:

```
proc freq data = sdtm.jl;
    tables lbtestcd*lbspec*lborres;
 run;
```

The results from the PROC FREQ will be used to get all values for the where clause condition variables LBTESTCD and LBSPEC which describe the LBORRES values.

ID	Dataset	Variable	Comparator	Value
LB.LBTESTCD.GLUC.LBSPEC.BLOOD	LB	LBTESTCD	EQ	GLUC
LB.LBTESTCD.GLUC.LBSPEC.BLOOD	LB	LBSPEC	EQ	BLOOD

Table 1 Worksheet entries for where clause LBTESTCD EQ "GLUC" AND LBSPEC EQ "BLOOD"

For each where clause an ID has to be defined which contains the concatenated names and values of the condition variables, according to best practice.

For each variable used in the where clause a record has to be created, the where clause ID remains the same.

5.2.4 Worksheet “ValueLevel”

The “ValueLevel” worksheet contains the value level metadata defined by where clause conditions.

On value level different data types can be specified for one variable: The character variable TS.TSVAL contains only date info with ISO8601 format where TSPARMCD = “DCUTDTC“ (Data Cutoff Date).

For TSPARMCD = “DOSE“ (Dose per Administration), TSVAL contains only integer values.

Where Condition	Label / Description	Type	Length or Display Format	Controlled Terms or ISO Format
TSPARMCD = "DCUTDTC" (Data Cutoff Date)	Parameter Value	datetime		ISO 8601
TSPARMCD = "DOSE" (Dose per Administration)	Parameter Value	integer	8	

Display 6 Value level metadata for TS.TSVAL

The value level metadata will be updated as follows:

Records from unchanged domains remain untouched. For new/updated domains new records will be created for each where clause ID previously defined. Data type and length will be derived from the variable values when restricted to the where clause condition.

Similar to the “Variables” worksheet the info for the columns “Format”, “Codelist”, “Origin”, “Pages”, “Method” and “Comment” can also be obtained from the existing define.xml. If this is not possible the columns have to be filled manually in the program or by rules that can be defined in an additional Excel sheet.

5.2.5 Worksheet “Codelists”

Codelists are defining all possible values for a variable. They can be originating from the CDISC terminology or be user-defined. Codelists can be assigned on variable or value level.

Unit (LBRESU) [*C71620*]

Permitted Value (Code)
% [*C25613*]
X10^9/L [*]
g/dL [*C64783*]
mg/dL [*C67015*]
ng/dL [*C67326*]
pg/mL [*]

* Extended Value

Display 7 Codelist with NCI codes and user-defined extended values

In general, the existing codelist sheet is used as base for updates.

Codelists which are linked to variables deleted by the update process will also be deleted by the program.

Exception: The corresponding codelist will only be deleted if not used by any other variable.

New codelists must be defined if required by new variables or new entries on value level. These new codelists can be either based on already existing codelists or must be created in the program.

If a codelist is following a CDISC terminology, the corresponding NCI codes have to be merged from the CDISC terminology file. The worksheet column „Term“ is derived from „CDISC Submission Value“.

Code	Codelist Code	Codelist Extensible (Yes/No)	Codelist Name	CDISC Submission Value	CDISC Synonym(s)
C64783	C71620		Unit	g/dL	Gram per Deciliter; g%

Display 8 CDISC SDTM terminology

Duplicate codelist entries might be created by adding new entries on variable or value level during the update process but are detected and deleted by the SAS program in a final cleanup step.

5.2.6 Worksheet “Methods”

Methods linked to variables or values which are removed by the update process are also removed from the “Methods” worksheet.

New methods must be added if required by new/updated variables and values.

5.2.7 Worksheet “Comments”

Same as for “Methods”.

5.3 SPECIFICATION EXPORT

The updated metadata info is exported via PROC EXPORT to Excel.

The new Excel specifications are used to create a new define.xml with Pinnacle 21 Community.

6 CONCLUSION

Updating a define.xml with an automatic process can be a convenient solution.

An important advantage is reproducibility, as changes in the source data or validation findings at the process end might make it necessary to run the process more than once. Manual Excel sheet updates, which can be error-prone, are avoided.

As prerequisite, the number of changes in the SAS data files should be limited. For new domains, metadata from already existing, similar domains should be available as a starting point.

Especially if the updates have to be done for various studies with a similar design an automatic process can save time and reduce the workload.

7 CONTACT INFORMATION

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

Steffen Müller
mainanalytics GmbH
steffen.mueller@mainanalytics.de
www.mainanalytics.de