A Practical Approach to Creating Define.XML by Using SDTM Specifications and Excel functions

Amos Shu
Endo Pharmaceuticals., Chadds Ford, PA 19317

Introduction

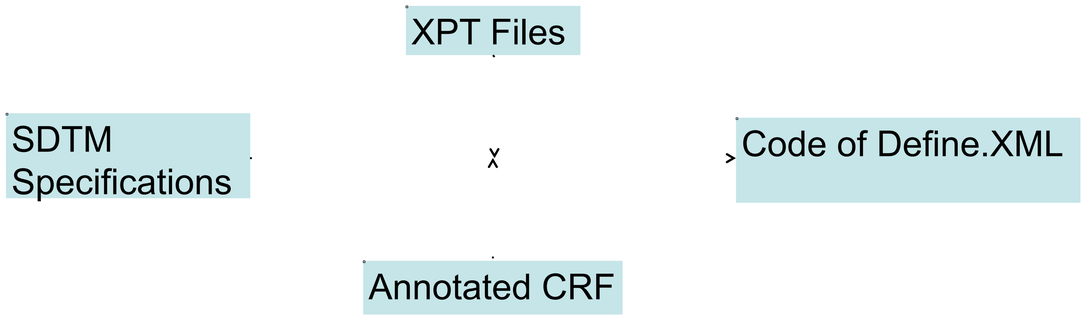
Define.xml (Case Report Tabulation Data Definition Specification) is a document that FDA required for drug submission. It describes the structure and contents of the data collected during the clinical trial process. Because Define.xml can increase the level of automation and improve the efficiency of the Regulatory Review process, FDA likes to have it with drug submission. The define.xml standard is based on the CDISC Operational Data Model (ODM), which is available at http://www.cdisc.org/standards/index.html.

To generate the code for Define.xml, there are three challenges [1] that average SAS programmers need to overcome:

1. Basic understanding of XML
2. Thorough understanding of the CDISC-specific XML structure of Define.xml
3. SAS expertise to generate the XML code

The first two challenges are fundamental; there are no alternatives or shortcuts to them. However, there are alternatives to the third one. Instead of SAS or XML tools, SDTM specifications and Microsoft Excel can be used to program Define.xml in a practical and efficient way.

Process Flow of Define.xml Code Generation



STEP 1. XPT File Generation

Before generating the code for Define.xml, first transform the SDTM datasets into .xpt files. SAS XPORT engine is designed to do this type of job. Either DATA-SET step or PROC COPY can be used to do this.

STEP 2. Annotated CRF Generation

An annotated CRF is usually available in most clinical trials, which is prepared by the data management team for collecting clinical trial data. The issue is that many variable attributes are modified across all SDTM datasets based on the SDTM Specifications, which vary with the specific statistical analysis plan (SAP). Those changes need to be added to the annotated CRF for Define.xml.

STEP 3. Use SDTM Specifications to Generate Code of Define.xml

Define.xml has four sections in general: 1. Table of Contents (TOC, or Data Metadata), 2. Collection of Data Definition Tables (Variable Level Metadata), 3. Controlled Terminology, and 4. ODM XML Header, Study, and MetaDataVersion. The first two sections are the main part of Define.xml.

1. Generate the TOC Section

The TOC lists all of the datasets (domains) included in the drug submission. It would be straightforward to create the following Excel sheet for TOC, based on the SDTM specifications and the SDTM IG. The last column will generate a hyperlink with the XPT files created earlier. Based on this sheet, you can use an ODM (Operational Data Model) element – **ItemGroupDef** to generate XML code for the TOC section.

Dataset	Description	Class	Structure	Purpose	Keys	Location
AE	Adverse Events Dataset	Events	One record per adverse event per subject	Tabulation	STUDYID, USUBJID, AEDECOD, AESTDTC	ae.xpt
…	…	…	…	…	…	…

2. Generate the variable level Metadata Section

It would not be difficult to use the SDTM specifications and follow the SDTM IG to create an Excel sheet like the following one for all domains:

Dataset Name	Dataset Label	Variable Number	Variable Name	Mandatory
AE	Adverse Events	1	STUDYID	Yes
…	…	…	…	…

The XML code for the first part of variable level metadata would look like this using **ItemRef** and Excel CONCATENATE function.

```
<ItemRef ItemOID="AE.STUDYID" OrderNumber ="1" Mandatory ="Yes" />
<ItemRef ItemOID="AE.DOMAIN"  OrderNumber ="2" Mandatory ="Yes" />
<ItemRef ItemOID="AE.USUBJID" OrderNumber ="3" Mandatory ="Yes" />
… …
```

The second Excel sheet looks like this:

Variable Name	Variable Label	Variable Type	Variable Length	Controlled Terms or Format	Origin	Comments	Computation
USUBJID	Unique Subject Identifier	text	18		Derived	STUDYID-SITEID-SUBJID	
…	…	…	…	…	…	…	…

ItemDef and Excel functions such as CONCATENATE and IF are used to generate the XML code for the second part of variable level metadata.

3. Controlled terminology (Code Lists) Section

This section is for variables that have a discrete list of valid values or controlled terms associated with them. For example, the route of dosing in the dataset of concomitant medications (CM) would have the following values:

Name	Description
IV	Intravenous
IM	Intramuscular
PO	Per oral
SC	Subcutaneous
PR	Per rectal
SL	Sublingual
INH	Inhaled
TOP	Topical
TD	Transdermal
TB	Transbuccal
Other	Other, specify
NMT	Non-Medicinal Therapy
…	…

The ODM **CodeList** element is used to generate XML code for the controlled terminology section.

4. ODM XML Header, Study, and MetaDataVersion Section

The document of Define.xml has good examples for creating this section. It can be easily completed by following the instructions.

Conclusion

The process of using SDTM specifications and Excel functions to generate the code of Defiune.xml is an easy, straightforward, and time-saving alternative with no additional cost. Average SAS programmers should not need extensive training to complete this task.

Reference

[1]. Molter, Michael, *A SAS® Programmer's Guide to Generating Define.xml*, SAS Global Forum 2009
[2]. Kawohl, Monika, *A SAS based solution for define.xml*, PharmaSUG, 2007
[3]. Adams, John, etc., *Creating a define.xml file for ADaM and SDTM*, PharmaSUG, 2011
[4]. Banga, Rohit, Generate Define.xml & Define.pdf from Metadata Environment, PharmaSUG, 2009
[5]. Becker, Matt, etc., *SDTM, ADaM and define.xml with OpenCDISC*, PharmaSUG, 2011
[6]. CDISC Define.xml, V 1.0.0 (http://www.cdisc.org)
[7]. http://www.cdisc.org/SDTM

EXAMPLE OF XML CODE

1. XML CODE FOR TOP SECTION

```
<ITEMGROUPDEF  OID="AE"
        NAME="AE"
        REPEATING="YES"
        ISREFERENCEDATA="NO"
        PURPOSE=" TABULATION"
        DEF:LABEL="ADVERSE EVENTS "
        DEF:STRUCTURE="ONE RECORD PER ADVERSE EVENT PER SUBJECT"
        DEF:DOMAINKEYS="STUDYID, USUBJID, AEDECOD, AESTDTC"
        DEF:CLASS="EVENTS"
        DEF:ARCHIVELOCATIONID="LOCATION.AE">

        … …

        <DEF:LEAF ID="LOCATION.AE"
                        XLINK:HREF="AE.XPT">
                        <DEF:TITLE>AE.XPT</DEF:TITLE>
        </DEF:LEAF>
</ITEMGROUPDEF >
```

2. XML CODE FOR THE VARIABLE LEVEL METADATA SECTION

```
<ITEMREF ITEMOID="AE.STUDYID" ORDERNUMBER ="1" MANDATORY ="YES" />
<ITEMREF ITEMOID="AE.DOMAIN"  ORDERNUMBER ="2" MANDATORY ="YES" />
<ITEMREF ITEMOID="AE.USUBJID" ORDERNUMBER ="3" MANDATORY ="YES" />
  … …

<ITEMDEF OID="AE.USUBJID"
        NAME="USUBJID"
        DATATYPE ="TEXT"
        LENGTH ="18"
        ORIGIN ="DERIVED"
        COMMENT =" STUDYID-SITEID-SUBJID "
        DEF:LABEL ="UNIQUE SUBJECT IDENTIFIER"
        DEF:DISPLAYFORMAT ="$18.">
        … …
 </ITEMDEF>
```

3. XML CODE FOR CONTROLLED TERMINOLOGY (CODE LISTS) SECTION

```
<CODELIST OID="CL.ROUTE" NAME="ROUTE" DATATYPE="TEXT">
                <CODELISTITEM CODEDVALUE="IV" DEF:RANK="1">
                                <DECODE>
                                        <TRANSLATEDTEXT XML:LANG="EN">INTRAVENOUS</TRANSLATEDTEXT>
                                </DECODE>
                </CODELISTITEM>
                <CODELISTITEM CODEDVALUE="IM" DEF:RANK="2">
                                <DECODE>
                                        <TRANSLATEDTEXT XML:LANG="EN">INTRAMUSCULAR</TRANSLATEDTEXT>
                                </DECODE>
                </CODELISTITEM>
                 … …
  </CODELIST>
```

The example of output of TOP section:

Dataset	Description	Structure	Purpose	Keys	Location
AE	Adverse Events	One record per adverse event per subject	Tabulation	STUDYID, USUBJID, AEDECOD, AESTDTC	ae.xpt

Two hyperlinks – Adverse Events and ae.xpt are created, which directly link to the corresponding variable level Metadata section and the xpt file of the specific domain, respectively.

The output of variable level metadata looks like the following:

Variable	Label	Type	Controlled Terminology	Origin	Comment
STUDYID	Study Identifier	text		Derived	
DOMAIN	Domain Abbreviation	text	AE	Derived	
USUBJID	Unique Subject Identifier	text		Derived	STUDYID-SITEID-SUBJID
AESTDY	Study Day of Start of Adverse Event	Integer		Derived	See Computational Method: AESTDY
AETERM	Reported Term for the Adverse Event	text		CRF Page 53	
AEDECOD	Dictionary Derived Term	text	MedDRA	Derived	

Shu