

Robust tools to Create Define.xml v2.0 based submission components for SDTM, ADAM & SEND

Vineet Jain, Independent Consultant

In 2013 CDISC came out with define XML specification version 2.0. Unlike previous version 1.0, which was focused on SDTM, version 2.0 define.xml is designed equally well for SDTM, SEND and ADaM. Define.xml version 2.0 is now integrated with NCI controlled terminology and supports complicated Value-level metadata definitions. Previous define.xml tools for define.xml V1.0 are incompatible with newer define version and more systematic and sophisticated tools are needed to comply with define.xml V 2.0 specifications.

This document, along with attached programs, presents unified metadata system to create define.xml version 2.0 for SDTM, ADAM or SEND data. Also additional tools are presented to use the same system to assist CRF annotation & create define.pdf. The system uses four standard metadata tables - one each for dataset, variables, value level and controlled terminology. The define.xml creation is split into two steps – creation of validated metadata datasets and using these to create define.xml. With use of validated metadata tables, define.xml creation becomes lean and issue-free. The detailed metadata specs are presented in the appendix.

Macros needed by the system along with few other files are attached in this pdf file, which the users are free to use and modify. Contact me to get support and/or customize the system for specific needs. To access the attached files via Adobe reader, select 'View' menu, select 'Show/Hide' -> 'Navigation Panes' -> 'Attachments'. To access the files in Foxit pdf software, open then attachments section from navigation pane on the left.

Introduction

The key features of presented system are:

- Generic define creation system for SDTM, ADaM & SEND (SEND not tested yet, but should work with minor tweaks)
- Create compliant **Define.xml v2.0** every time, with use of inbuilt checks
- Relieves users from editing/inspecting the xml and understanding its specification
- Assist in **CRF annotation** consistent with define.xml
- Create identical **define.pdf** from define.xml v2.0 along with links & bookmarks; any externally generated define.xml v2.0 can be used. Define.pdf created with the help of Apache FOP software.
- Tools based on light-weight & customizable SAS macros; these can be easily run in any SAS environment & integrated with front end systems

Figure 1 (on next page) shows the workflow of the system. The process consists of two broad steps:

- Creation of validated metadata tables
- Use of these metadata tables to create define.xml & annotated CRF

Additionally, identical **define.pdf** can be created from define.xml with the tools discussed in this document later.

Metadata needed for Define.xml is usually created from multiple sources. Most of the study metadata can be extracted from SAS datasets, yet some metadata has to be separately created (manually via dataset specs, from CRF fields and/or by using metadata libraries) for define.xml. This includes dataset comments, variable

origin/comment/methods, Valuelevel metadata, controlled terminology, NCI codes for standard terminology and external dictionary details. The metadata consolidation method/process can vary across programming teams and across data standards (SDTM, ADAM and SEND), and it becomes hard to create generic metadata consolidation tool across teams. This document focusses on defining the standard metadata structure and its use to create define v2.0 based deliverables.

These metadata tables are discussed in detail in next section. The metadata specification & corresponding QC checks are presented in Appendix.

Standardized metadata datasets capture all the metadata needed for define.xml and greatly simplify error checking. With validated metadata, define.xml creation becomes painless & robust. Following tools use these metadata tables:

- **Metadata Validation** (define checks.sas): Run several checks to ensure compliant define deliverables are created
- **CRF annotation** (CRFanno.sas): Creates FDF file from metadata tables. The FDF file can be imported by pdf software (e.g. adobe/foxit) to fill in annotations. Refer section: [CRF Annotation](#).
- **Define.xml v2.0** (defxml.sas): Create complete & compliant define.xml. Refer Section: [Define.xml Generation](#) for details

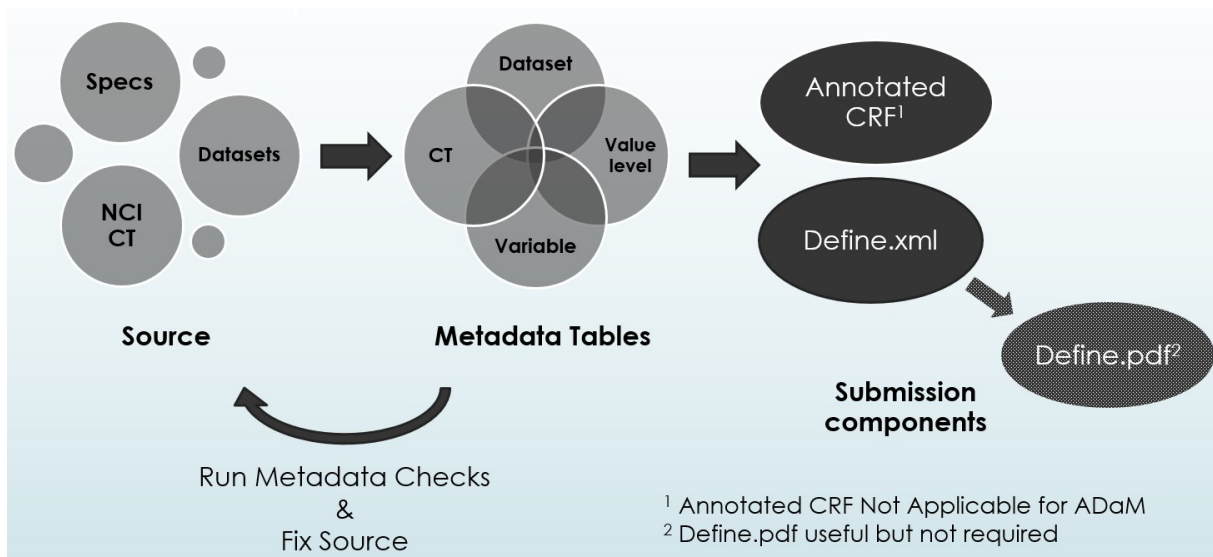


FIGURE 1: DEFINE CREATION WORKFLOW

Additionally users interested in creating **define.pdf** can use the attached XSL file to render the xml file to pdf via Apache FOP, a free open source software. The file can convert compliant define.xml v2.0 file (created by any tool either in this paper or other) to define.pdf. The define.pdf looks identical to define.xml (when viewed using XSL stylesheet from CDISC) and includes the internal/external links & bookmarks.

Metadata Tables

The system in this paper use following four metadata tables:

DEFDS – Dataset level Metadata, with one row per dataset

DEFVAR – Variable Level Metadata, with one row per variable per dataset

DEFVL – Value Level Metadata, with one row per value list definition per variable per dataset

DEFFMT – Format/controlled terminology/External dictionary metadata, with one row per coded value per format

[Table 1](#), [Table 2](#), [Table 3](#) & [Table 4](#) in next are sample ADAM metadata. Purpose & use of most of the fields is self-explanatory. Specifications for the metadata datasets is presented in [Appendix](#). For detailed understanding of specific fields/attributes please refer define.xml V2.0 specifications & sample define package at CDISC website.

The metadata tables & corresponding define.xml program are designed to provide thorough support of define.xml V2.0 specifications from CDISC, yet currently some of optional or rarely needed features are not supported. Note this is not an exhaustive list of features not supported.

- Provision is not made to specify SASFormatName for codelists, since custom SAS formats are not supposed to be present in the CDISC based datasets anyways.
- Current define.xml macro support external links only for reviewers guide, computational algorithms & blankCRF.
- RoleCodeListOID not implemented as it is optional and not printed in browser while viewing define.xml

Creating Metadata Tables

As discussed in introduction section, metadata creation tool cannot be generalized due to variations in dataset specification design, tools & processes used across teams. Users planning to use the system in this paper, should develop their own SAS macro (or other tool) to generate the metadata tables or contact me for support. This section lists data/meta sources needed to build such macro.

Metadata needed for define creation, comes from these sources:

- Study datasets (SDTM/ADaM/SEND): To pull datasets, variable lists with basic metadata (e.g. names, labels, lengths etc.) and derived metadata (e.g. significant digits, display format) for variable & value-level metadata
- Data Specifications: To get additional dataset metadata (e.g. structure, class, comment, and key variables), variable metadata (e.g. origin, comments, format name), list of ValueLevel metadata along with its meta (e.g. where clauses, format names, comments) and controlled terminology metadata
- NCI terminology: To get NCI codes where standard terminology is used for data values, test codes, test names etc.
- CDISC data standard IG: To get variable roles, mandatory values, NCI terminology names, etc.
- Sponsor specific metadata libraries: Sponsors may have their own common metadata libraries for company/therapeutic area/drug and one can use such libraries to populate common metadata.

Although the metadata tables look very tedious to generate, smartly designed metadata consolidation macro can derive majority of metadata, minimizing manual data entry & time spent on quality control.

Table 1: DEFDS Sample

DATASET	DOMAIN	LABEL	STRUCT	CLASS	REPEATING	ISREF	PURPOSE	COMMENT	ORDER	DOCREF1
ADSL		Subject Level Analysis Dataset	One record per each subject	SUBJECT LEVEL ANALYSIS DATASET	No	No	Analysis	The source SAS data sets for ADSL are the following subject-level data SDTM datasets: DM, DS, EX, MH, SC, SV, QS, and VS. Refer Reviewer's Guide Section 1.1	1	ReviewersGuide# ND#Section1.1
ADLB		Laboratory	One record per each subject, per parameter, per visit	BASIC DATA STRUCTURE	Yes	No	Analysis	Only keep randomized patients (ADSL.RANDFL = Y). Refer Reviewer's Guide Section 1.2	2	ReviewersGuide# ND#Section1.2
ADQS		Questionnaire	One record per each subject, per parameter, per visit.	BASIC DATA STRUCTURE	Yes	No	Analysis	Only keep randomized patients (ADSL.RANDFL = Y)	3	

Table 2: DEFVAR Sample

DATASET	VARIABLE	LABEL	LENGTH	ORDER	FMTNAME	ORIGIN	ORGDETL	MANDATORY	ROLE	DATATYPE	DISPFMT	SIGDIGIT	KEYSEQ	COMMENT	METHYP	DOCREF1
ADSL	STUDYID	Study Identifier	12	1		Predecessor	DM.STUDYID	Yes		text			1			
ADSL	USUBJID	Unique Subject Identifier	11	2		Predecessor	DM.USUBJID	Yes		text			2			
ADSL	SUBJID	Subject Identifier for the Study	4	3		Predecessor	DM.SUBJID	Yes		text						
ADSL	TRTP	ADaM Description of Planned Arm	20	18	ARM	Derived		No		text				Derived from DM.ARM	Computation	
ADSL	BMIBL	Baseline BMI	8	28		Derived		No		float		1		VS.STRESN where VSTESTCD=BMI and VISITNUM=1	Computation	

Table 3: DEFVL Sample

DATASET	VARIABLE	DATATYPE	LABEL	DISPFMT	SIGDIGIT	LENGTH	FMTNAME	ORIGIN	ORGDETL	COMMENT	METHYP	WHERE1	WHERE2	ORDER	DOCREF1
ADQS	AVAL	float		5.2	2	8	ACIT1	Derived		QS.QSSTRESN where QSTESTCD= PARAMCD		PARAMCD IN 'ACITM01', 'ACITM02', 'ACITM03'		1	
ADQS	AVAL	Integer		2		8		Derived		QS.QSSTRESN where QSTESTCD= PARAMCD		PARAMCD IN 'ACITM04', 'ACITM05', 'ACITM06', 'ACITM07', 'ACITM08', 'ACITM09', 'ACITM10', 'ACITM11', 'ACITM12', 'ACITM13', 'ACITM14'		2	
ADQS	AVAL	Integer		2		8		Derived		Sum of ADAS scores for items 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, and 14,	Computatio n	PARAMCD EQ 'ACTOT'	ANL01FL EQ 'Y'	3	
ADQS	CHG	float		5.2	2	8		Derived		AVAL – BASE	Computatio n	PARAMCD EQ 'ACTOT'		1	

Table 4: DEFFMT Sample

FMTNAME	FMTLAB	FMTTYPE	VALUE	DECODE	NCIFMT	NCITEM	ORDER	DATATYPE	DICTNM	DICTVER	RANK
AGEGRP	Age Group	CT	<65				1	text			1
AGEGRP	Age Group	CT	65-80				2	text			2
AGEGRP	Age Group	CT	>80				3	text			3
YNONLY	No Yes Response	FORMAT	N	No	C66742	C49487		text			
YNONLY	No Yes Response	FORMAT	Y	Yes	C66742	C49488		text			
YONLY_N	Yes Response (N)	FORMAT	1	Yes				integer			
AEDICT	AE Dictionary	DICT						text	MEDDRA	15.0	

Metadata Validation

All four metadata tables must be validated before passing those to define.xml in order to create a compliant define.xml and avoid the define.xml creation macro to fail. If the metadata datasets aren't fully validated then the macro may still successfully create define.xml, but the xml file may have issues. It is hard to visually find all the issues that may be present in define.xml.

Following is list of key validation checks those must be performed on metadata datasets before using the define.xml program:

- Metadata tables are well formed
 - All the needed metadata variables with correct data type are present
 - Variables values are valid
 - Variables must have non-missing value where they are required or conditionally required
 - Non-printable characters are not used in any of character fields
 - Variable value must exactly match (including case) one of possible values from value list, where list of allowed values is specified
 - ORGDETL, DOCREF1, DOCREF2... & WHERE1, WHERE2... values are machine readable
 - Case of variables, DATASET, VARIABLE, FMTNAME are in uppercase to ensure successful merge between datasets
- Metadata tables are consistent With study datasets and CDISC/NCI terminology/codes
 - Metadata accurately represent the contents of datasets
 - Variable list match the SAS proc contents output
 - For variables & value-level metadata, if a format/CT is defined, the Controlled terminology must include all values present in corresponding data
 - NCI CT correctly used wherever applicable
 - DEFVAR.KEYSEQ is defined for one or more variables for each dataset and dataset sorted using the same key variables
- Metadata tables are consistent with each other
 - References to a CT (in variable/ValueLevel metadata) not missing in CT metadata
 - Value level data attributes consistent/compatible with those of parent variable definition. I.e. origin/ DataType/length are consistent. For example, no value-level length can exceed the length of the parent variable, (from Section 4.4 in CDISC define.xml Specs [1])
 - Format DataType matches with the DataType of variable using the format
 - Order, rank variables have unique sequential integer values within corresponding group of observations when non-missing.

These validation checks can be programmatically created and should be tested before the use of metadata tables.

Majority of above checks are performed by the program **define checks.sas**. The list of checks performed the program are specified in [Appendix](#).

Define.xml Generation

Once the validated metadata datasets are ready, attached macro defxml.sas can create compliant define.xml quickly & painlessly without the need to understand & worry about XML structure. This macro currently works for SDTM & ADaM based data. With few minor modifications it should work for SEND also, since define v2.0 specifications has SEND designed very similar to SDTM.

The macro reads the metadata tables in form of SAS datasets to create define.xml. The program starts with cleaning up some of the special characters which are not permitted in define.xml. It then creates define.xml header and assembles the define.xml section by section using the four metadata datasets.

Refer the program header to know more about the paths & parameters needed for the program.

The macro assumes the metadata datasets are already QCed and are in sync. So it doesn't have built-in checks or imputations and creates define.xml quickly in few seconds.

Define.xml QC

With use of validated metadata tables, the above created define.xml is not expected to have any structural/compliance related issues. Yet it is important to perform thorough QC on define.xml, until metadata checks are mature. Following techniques are recommended to ensure define.xml is compliant & meets quality:

- Visual QC: to primarily review metadata contents and external links; finding compliance related problems can be very tedious with visual check and it can't be relied upon to ensure define.xml compliance.
- OpenCDISC xml Checks: Performs several compliance checks, very useful to check define.xml against data
- Validate against Schema: To check if xml file adheres to the schema by CDISC. Several ways are possible to perform this check, refer <http://www.cdisc.org/define-xml> for more details.

If any structural/compliance related issues are found in above checks, corresponding checks should be added to programmatic checks for metadata tables to ensure these are caught upfront. Robust programmatic checks ensure define.xml is free of all known issues.

Define.pdf Generation

Define.pdf can be a great supplement to define.xml for multiple reasons.

Using the attached xsl file – xml2fo.xsl -, Apache FOP software (<http://xmlgraphics.apache.org/fop/>) can render the define.xml to define.pdf. The generated define.pdf looks almost identical to corresponding define.xml, viewed via standard stylesheet provided by CDISC in define v2.0 package. It is because, xml2fo.xsl is created using the same CDISC's stylesheet: define2-0-0.xsl. CDISC's stylesheet render xml to xhtml, whereas xml2fo.xsl renders xml file to 'fo' format. FO (formatting objects) is an intermediate file format and users don't need to dig into its technical details since, Apache FOP can directly use this xsl file to render xml to pdf.

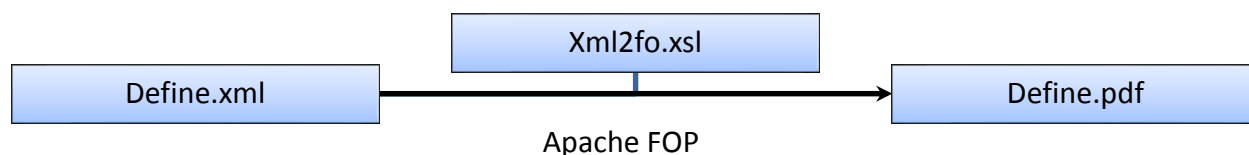


FIGURE 2: DEFINE.PDF GENERATION

Apache FOP is an open-source free Java application that converts XML and XSL Formatting Objects (XSL-FO) files to PDF or other printable formats. It can be downloaded from <http://xmlgraphics.apache.org/fop/>. It successfully works in Microsoft Windows. Since it is a java application it is expected to work in unix as well.

Define.pdf generation using xml2fo.xsl is independent of metadata tables presented in this document. Any version 2.0 compliant define.xml can be used to create corresponding define.pdf using presented technique.

To use apache FOP, download apache FOP software and use following command (Replace ... with full file paths) in fop installation directory (or create an alias to use the application from any directory) to create define.pdf from define.xml.

```
fop -xml ...define.xml -xsl ...xml2fo.xsl -pdf ...define.pdf
```

It is important that the define.xml is free of structural/compliance issues, otherwise Apache FOP would fail to properly create define.pdf. If any problem exists in the xml file, Apache FOP creates helpful messages in command window to point towards the source of the problem. Make sure to scan the messages and make necessary fixes. One great feature of apache FOP is that it checks all the internal links while creating pdf and let user know if any of the links are not working.

After define.pdf is ready, ensure external links (e.g. to CRF pages) point to correct pages, by testing a few links.

Define.pdf generation using apache FOP do has few limitations that users must keep in mind:

- Apache FOP can't wrap non-hyphenated words longer than the cell-width and truncates those. The current cell width chosen in xml2fo.xsl would make such cases rare, yet Apache notifies the user via messages whenever it happens. To fix such issues, either the problematic word must be shortened or hyphenated (use '-' in the word where it needs to split). Alternatively the cell widths (in xml2fo.xsl) can be adjusted if modifying the metadata is not an option.
- Apache FOP doesn't support creation of links to external documents (such as CRF, reviewers guide) within the bookmarks. Define.xml has links for blank crf & reviewer's guide. These links are generally useful in submission directories and if these are needed in define.pdf, one can manually insert these, within minutes, using professional pdf softwares.

CRF Annotation

With origin already populated in metadata tables, attached utility macro crfanno.sas can be used to expedite CRF annotation while keeping the annotated crf consistent with define.xml. The macro reads metadata tables to generate an FDF file with annotations, which can then be imported into blank CRF by pdf softwares. Refer the attached program to understand the use of the macro. Annotations created by the macro have been successfully imported in Adobe as comments.

To know more about FDF files refer papers already published in pharmaSUG about use of FDF for CRF annotations (e.g. *Automating Production of the blankcrf.pdf* by Walter Hufford, *PharmaSUG 2014*).

For SDTMs, the macro creates annotations along with following features:

- Automatic text: Appropriate text for domain name, variables, supp variables, testcds etc.
- Automatic font size: differentiate font size of annotation with domain name vs rest of annotations
- Automatic Text box color: To differentiate variables for multiple domains created on same crf page.
- Automatic Text box size: Calculate the text box size needed for the annotation text (for Arial font only).

With automatic determination of text, font & sizes CRF annotation becomes very quick and consistent. See [Figure 3](#) for sample annotations created by the Crfanno macro.

Sponsor: ABC Company	
Protocol: ABC-123	STUDYID
Subject:	SUBJID
Demographics	
DM = Demographics	SC = Subject Characteristics
Race	RACE <input type="radio"/> White <input type="radio"/> Other RACEOTH in SUPPDM
SEX	<input type="radio"/> Male <input type="radio"/> Female SEX
Initials	SCORRES when SCTESTCD=SUBJINIT
Informed Consent Date	____ YYYY/MM/DD RFICDTC
HEIGHT	____ cm VSORRES when VSTESTCD = HEIGHT
WEIGHT	____ Kg VSORRES when VSTESTCD = WEIGHT
VS = Vital Signs	

FIGURE 3: SAMPLE ANNOTATIONS CREATED USING FDF FILE OBTAINED FROM CRFANNO.SAS MACRO

After importing the FDF file into the blankcrf user still needs to reposition the text boxes to right location and check if any annotations need to be dropped/added. If any annotations are added/dropped same changes must be made to source metadata to ensure define.xml remains in sync with annotated CRF. Sometimes manual updates to annotations may be needed, e.g. if date & time for certain SDTM variable are collected in two separate fields located apart from each other on a CRF page, one may want to duplicate & modify the annotation from 'xxDTC' to 'Datepart of xxDTC' and 'Timepart of xxDTC'. Desired annotation for such cases can't be automatically generated with the metadata system presented here.

The annotation macro is easily customizable to change font, color & sizing schemes in order to meet specific needs.

Attachments

Following files are attached in this pdf file.

- Defxml.sas – SAS macro to create xml file from SAS metadata datasets
- Crfanno.sas – SAS macro to create FDF file with annotations
- define2-0.xsl – An enhanced version of CDISC's xsl stylesheet for define.xml with protocol number in bookmarks (replacing CDISC IG version) and suppress display of redundant comments/method section at the end of xml file.
- Xsl2fo.xsl – A xsl file used by Apache FOP to convert define.xml to define.pdf. This file has same enhancements to define.pdf as are in above xsl file define2-0.xsl
- Maxlen.sas – A utility SAS macro used often by other attached SAS macros to adjust the lengths of common variables when multiple datasets are merged with varying lengths of common variables.

References

1. CDISC Define-XML Specification, Version 2.0 at <http://www.cdisc.org/define-xml>

Contact Information

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

Name: Vineet Jain

E-mail: Vineet7878@gmail.com

Web: www.linkedin.com/in/vineet7878/

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Disclaimer

ALL CODE ATTACHED WITH THE FILE IS FREE FOR YOU TO USE OR MODIFY IN ANYWAY YOU LIKE. IF YOU HAVE ANY ADDITIONS OR REQUEST, FEEL FREE TO CONTACT ME. PLEASE REMEMBER THAT ATTACHED CODE MAY NEED SPONSOR/PROJECT SPECIFIC CUSTOMIZATIONS AND WOULD NEED TO BE THOROUGHLY TESTED BEFORE BROAD PRODUCTION USE. IT MAY CONTAIN ERRORS. IF YOU CHOOSE TO USE ANY OF THE ATTACHED CODE, OR PARTS OF IT, YOU DO SO ENTIRELY AT YOUR OWN RISK.

THIS SOFTWARE IS PROVIDED BY ME (VINEET JAIN) "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ATTACHED CODE IS COPYRIGHTED, AND I (VINEET JAIN) RETAIN SOLE RIGHTS TO IT. IT IS NOT TO BE REDISTRIBUTED FOR PROFIT OR BUNDLED WITH ANY OTHER PRODUCT WITHOUT WRITTEN CONSENT.

Appendix

Table 5: DEFDS dataset Specifications

Name	Purpose, Notes & Example	Required	Allowed values	Define.xml Attribute Name
DATASET	Name of Dataset, e.g. DM, ADSL	Y		Name, SASDatasetName
DOMAIN	Same as dataset name (except for split domain, where it is the name of root domain) Not applicable for ADAM. E.g. DM, AE, SUPPAE	Y (SDTM & SEND only)		Domain
REPEATING	Indicates whether domain can have more than one record per subject, e.g. No for DM & ADSL; Yes for AE, ADAE & SUPPAE	Y	Yes, No	Repeating
ISREF	Indicates whether the dataset contains reference data, e.g. Yes for TA, TE, TI and No for DM, ADSL, LB, ADLB.		Yes, No	IsReferenceData
PURPOSE	Purpose of Dataset, Value must be Tabulation for SDTM & SEND datasets whereas Analysis for ADAM datasets	Y	Tabulation, Analysis	Purpose
STRUCT	Structure (level of detail) of dataset, e.g. One record per subject for DM, ADSL, and 'One record per adverse event per subject' for AE, ADAE	Y		def:Structure
CLASS	Class of Dataset, e.g. BASIC DATA STRUCTURE for ADLB	Y	For SDTM & SEND (SPECIAL PURPOSE, FINDINGS, EVENTS, INTERVENTIONS, TRIAL DESIGN, RELATIONSHIP) For ADAM (SUBJECT LEVEL ANALYSIS DATASET, BASIC DATA STRUCTURE, ADAM OTHER)	Def:Class
LABEL	Short Description of Dataset, E.g. Laboratory for ADLB	Y		Description
ORDER	Numeric Order in which Dataset must be displayed, E.g. ORDER = 1 for ADSL usually	Y		-
COMMENT	Dataset Documentation, it can be a long multi-line text description. For really long multi-paragraph description use Reviewer's Guide instead.			Description in def:CommentDef
*DOCREF1, DOCREF ...	External Document Reference, refer end of Appendix for details			def:DocumentRef for def:CommentDef

Table 6: DEFVAR dataset Specifications

Name	Purpose & Notes	Required	Allowed values	Define.xml Attribute
DATASET	Name of Dataset	Y		ItemGroupDef/ Name
VARIABLE	Variable Name	Y		ItemDef/ Name
LABEL	Variable Label	Y		ItemDef/ Description
FMTNAME	Controlled terminology/format name			ItemDef/ CodeListRef/ OID & Name
DATATYPE	Data type for Variable, e.g. text for USUBJID	Y	Refer section 4.2.1 in define specs [1]	DataType
LENGTH	Numeric Variable length	Y (only when datatype is 'text', 'integer' or 'float')		length
SIGDIGIT	Numeric Significant decimal digits	Y (only when datatype = 'float')		SignificantDigits

Name	Purpose & Notes	Required	Allowed values	Define.xml Attribute
DISPFMT	data format for visualization of numeric float and date Values. E.g. 8.2 and date9.			Def:DisplayFormat
ORIGIN	Variable Origin	Y	CRF, Derived, Assigned, Protocol, eDT, Predecessor	def:Origin/ Type
ORGDETL	Origin Detail. Must be populated when ORIGIN is CRF or Predecessor. When ORIGIN is CRF, ORGDETL should have page numbers separated by spaces (no other delimiters permitted). When Origin is Predecessor, ORGDETL should specify the source. Usually it would be datasetName.VariableName E.g. For variable USUBJID in ADSL, ORGDETL can be DM.USUBJID.	Y when ORIGIN is CRF or Predecessor		def:Origin/ ...
KEYSEQ	Numeric variable to indicate order of variable in key variables for the dataset, e.g. if key variables are STUDYID & USUBJID for a dataset then KEYSEQ must have value 1 for STUDYID and 2 for USUBJID within the dataset	Y for Key variables in the dataset	1, 2, 3...	KeySequence
MANDATORY	Indicates whether Variable is required in the dataset or not. Specification describe its use for SDTM only and it is not clear how the field must be used for ADAM.	Y	Yes, No	Mandatory
ROLE	Variable Role. Optional for SDTM standard domains. The values provided by the SDTM or SEND IGs are used for the standard domains. Conditional required for SDTM custom domains Not applicable for ADaM.	Conditional	Refer SDTM/SEND IG	Role
ORDER	Numeric Variable order number in the dataset	Y		OrderNumber
COMMENT	Variable comment or Derivation			def:CommentDef or MethodDef/ Description
METHYTP	Method Type when Origin = Derived	Y (when Origin= Derived)	Computation Imputation	MethodDef/ Type
*DOCREF1, DOCREF...	External Document Reference, refer end of Appendix for details			def:CommentDef/ def:DocumentRef

Table 7: DEFVL dataset Specifications

Name	Purpose & Notes	Required	Allowed values	Define.xml Attribute
DATASET	Name of Dataset	Y		ItemGroupDef/ Name
VARIABLE	Variable Name for which the Value List Definition is associated	Y		ItemDef/ Name
LABEL	Value Label, assign QLABEL for QVAL metatdata only (in SDTM SUPPQUAL datasets)	Conditional	QLABEL value from SUPP dataset	Description
**WHERE1, WHERE2...	Where clause, refer end of Appendix for details	Y		def:WhereClauseDef/ RangeCheck/ def:ItemOID, Comparator & CheckValue
FMTNAME	Controlled terminology/format name			CodeListRef/ OID & Name
DATATYPE	Data type for Variable	Y	Refer section 4.2.1 in define specs [1]	DataType
LENGTH	Numeric Variable length	Y (only when datatype is 'text', 'integer' or 'float')		length
SIGDIGIT	Numeric Significant decimal digits	Y (only when datatype = 'float')		SignificantDigits
DISPFMT	Data format for visualization of numeric float and date values. E.g. 8.2 and date9.			Def:DisplayFormat

Name	Purpose & Notes	Required	Allowed values	Define.xml Attribute
ORIGIN	Values Origin	Y	CRF, Derived, Assigned, Protocol, eDT, Predecessor	def:Origin/ Type
ORGDETL	Origin Detail. Must be populated when ORIGIN is CRF or Predecessor. When ORIGIN is CRF, ORGDETL should have page numbers separated by spaces (no other delimiters permitted). When Origin is Predecessor, ORGDETL should specify the source. Usually it would be datasetName.VariableName E.g. For variable USUBJID in ADSL, ORGDETL can be DM.USUBJID.	Y when ORIGIN is CRF or Predecessor		def:Origin/ ...
ORDER	Numeric order of value level metadata with a Variable. Can be skipped if order has no significance			OrderNumber
COMMENT	Comment			def:CommentDef or MethodDef/ Description
METH_TYP	Method Type when Origin = Derived	Y (when Origin= Derived)	Computation Imputation	MethodDef/ Type
*DOCREF1, DOCREF2...	External Document Reference, refer end of Appendix for details			def:CommentDef/ def:DocumentRef

Table 8: DEFFMT dataset Specifications

Name	Purpose & Notes	Required	Allowed values	Define.xml Attribute
FMTNAME	Controlled terminology/format name	Y		CodeListRef/ OID & Name
FMTLAB	Format label/name. For NCI/CDISC Controlled Terminology, this must exactly match the CodeList Name. Otherwise it can have short label or just copy FMTNAME for ease.	Y		CodeList/Name
FMTTYPE	Classify format as Format (i.e. those with code/decode pair), CT (for formats with coded value only) or DICT (for external Dictionaries)	Y	FORMAT CT DICT	
DATATYPE	Data type for format, must match with data type for variables using the format	Y	text float integer	CodeList/ DataType
VALUE	Coded value text	Y		CodedValue
DECODE	Decode for coded value			Decode
ORDER	Numeric display order for values within FMTNAME, can be skipped if order has no significance			OrderNumber
RANK	Numeric Significance of coded value			Rank
NCIFMT	NCI codelist code for the format e.g. C66767, applicable only when FMTTYPE=FORMAT			CodeList/ Alias/ Name
NCIITEM	NCI item code for the format e.g. C66767, applicable only when NCIFMT is present			EnumeratedItem/ Alias/ Name
DICTNM	Dictionary Name	Y (when FMTTYPE = DICT)		ExternalCodeList/ Dictionary
DICTVER	Dictionary Version	Y (when FMTTYPE = DICT)		ExternalCodeList/ Version

***Document Reference Variables - DOCREF1, DOCREF2...**

Document Reference variables are used to insert hyperlinks to external documents such as reviewer's guide. Every external document, that needs to be linked, needs a document code to refer to it. Current define system supports three document reference codes: Blankcrf, ReviewersGuide & ComplexAlgorithms.

A document reference can be just link to just the external document or it can be link to specific location within the external document. If the reference to just the document is needed, then assign the document code to DOCREF# variable. E.g. DOCREF1= ReviewersGuide. If the reference needs to point to a specific location within the target document then document reference needs destination type & destination name along with the document code, separated by delimiter '#'. E.g. ReviewersGuide#ND#Section1.1, where ND is destination type & section 1.1 is actual destination within Reviewer's guide.

Three types of destinations are possible:

- ND: Named Destination – E.g. ReviewersGuide#ND#Section1.1. With named destination type link, one can hyperlink to specific named destination (such as sections, tables, figures) within external document. The link works only if the named destinations exist within the external document. Refer pdf documentation to understand how to create named destinations in pdf files
- PR: Page reference – E.g. ReviewersGuide#PR#5. This option is easier to use and links to a specific page in external document
- PRR: Page Range Reference – E.g ReviewersGuide#PR#5 7. When one needs to refer to a range of pages, Page range reference type destination should be used. With this option specify starting page & ending page numbers, separated by space in the destination name.

More than one reference/hyperlink can be inserted within a comment by having one DOCREF# just variable per hyperlink. While using multiple document references, always insert first document reference to DOCREF1, second to DOCREF2 to make sure define.xml doesn't miss any references. Also, these variables should be ordered in SAS dataset left to right as DOCREF1, DOCREF2... to ensure define.xml inserts the references in same order in the comments.

****Where Clause Variables - WHERE1, WHERE2...**

Where Clauses specify the condition when specific definition/algorithm in Valuelevel metadata is applied. These where conditions are machine readable and need to follow specific syntax. Each where clause must have three parts - Variable name, comparator and Variable value<s> separated by spaces. Variable name can be any variable in current dataset. Comparator can be 'LT', 'LE', 'GT', 'GE', 'EQ', 'NE', 'IN' or 'NOTIN'. Values need to be quoted and separated by either spaces or ','.

More than one where clauses can be part of single value level metadata instance. E.g. the condition *LBTESTCD EQ 'BILI' AND LBCAT EQ 'CHEMISTRY' AND LBSPEC EQ 'BLOOD'* has three where clauses. Each of the where clauses must be defined in separate 'WHERE#' variable with variable names are WHERE1, WHERE2, WHERE3... While using multiple where clauses, always insert first where clause in WHERE1, second in WHERE2 and so on to make sure define.xml doesn't miss any WHERE clause. Also, these variables should be ordered in SAS dataset left to right as WHERE1, WHERE2... to ensure define.xml has the combined where clause in same order.

Table 9: List of checks in define checks.sas

ERRCD	Description
DF001	Domain Metadata: Invalid value (missing, too long or not following CT)
DF002	Variable Metadata: Invalid value (missing, too long or not following CT)
DF003	Value Level Metadata: Invalid value (missing, too long or not following CT)
DF004	Format Metadata: Invalid value (missing, too long or not following CT)
DF005	Variable length invalid (Length must be valid & needed only when DataType is text, integer or float)
DF006	Invalid/missing Variable Origin/origin pages (Missing allowed only for variables those have Value level metadata e.g. QNAM, --ORRES)
DF007	Missing comment or method-type (METH TYP) when variable's origin is derived
DF008	Non-missing Variable method-type (METH TYP) where comment is missing
DF009	ValueLevel length invalid (Length must be valid & needed only when DataType is text, integer or float)
DF010	Missing comment or method-type (METH TYP) when ValueLevel origin is derived
DF011	Non-missing value-level Variable method-type (METH TYP) where comment is missing
DF012	Invalid/missing Value level Origin/origin pages
DF013	Non-Printable Characters in Dataset comment
DF014	Non-Printable Characters in Variable comment
DF015	Non-Printable Characters in Valuelevel comment
DF016	Dataset name missing in either domain list or Variable list for define.xml creation
DF017	Value-level metadata defined for a variable that is not present in variables metadata
DF018	DataType for Valuelevel metadata incompatible with DataType of corresponding variable
DF019	Origin for Valuelevel metadata incompatible with Origin of corresponding variable
DF020	Length/significant digits for Valuelevel metadata longer than the that for corresponding variable
DF021	No key Variable defined for dataset
DF022	Key variables not available in uninterrupted integer sequence (1, 2, 3..) for the dataset
DF023	Format for Valuelevel metadata different from the one used for corresponding variable (when both are non-missing)
DF024	Format used by multiple variables/value-level without common DataType
DF025	Format referred by variable/value-level but missing DEFFMT
DF026	Format present in DEFFMT but no variable/value-level data refers to the format
DF027	DataType for the format does not match for the one for variable/value-level using the format
DF028	Data value in source dataset not present in corresponding format/CT defined in DEFFMT
DF029	Data doesn't have any record satisfying where condition in define.xml for value-level metadata.
DF030	In Value-level metadata, where term can't be parsed successfully