



AUTOMATIC CREATION OF DEFINE.XML FOR ADAM: A FAST WAY APPROACH STARTING FROM ADAM METADATA

Dmitri Petratchenko, Valos, Genova, Italy

Andrea Parodi, Valos, Genova, Italy

Anna Romanova, Valos, Belarus

INTRODUCTION

The purpose of the define.xml is informing the reviewer (i.e. authority, CRO, external users) on what datasets, variables, controlled terms, and other specified metadata were used for building ADaM datasets.

Pinnacle 21 Community® creates define.xml starting from a detailed xls input file that is really time-consuming to be created. In this poster we describe a SAS® macro created by Valos that exploits the ADaM Metadata for speeding up the process of creation of this file (we will call it Excel Spec throughout this poster) through some instructions to be put in ADaM Metadata during its compilation.

ADAM METADATA

In the two examples below two sheets from ADaM Metadata are shown. The key columns holding the instructions for the macro are “Codelist”, “Origin” and “Source/Derivation”.

In the "Codelist" column the user must specify just the name of the codelist for the variable. This will be processed then by the macro as detailed then in this paper.

The “Origin” column with origin type information, can be filled with four different definitions:

- Predecessor: for variables pre-existing in an SDTM domain or in another ADaM dataset
- Assigned: for variables that are a classification of others like for example

Dataset Name	Parameter Identifier	Variable Name	Variable label	Variable Type	Display format	Codelist/Controlled Term	Source/Derivation	Comments	Codelist	Origin
ADSL	Perm	COUNTRY	Country	text	S3		DM.COUNTRY			Predecessor
ADSL	Perm	COUNTRYL	Country (Decode)	text	S50		One-to-one map to COUNTRY (according to C66786)		COUNTRY	Assigned
ADSL	Perm	COUNTRYN	Country (N)	integer	S		Uniquely linked to COUNTRYL, numeric. Coded in alphabetical order		COUNTRYN	Assigned

- Derived: for variables derived from pre-existing variables by means of a specific algorithm
- Composite: the origin column is filled using the below structure:

@<order number>@<origin>@<variable>@<comparator>@<condition>;

Where <order number> is a provisional index number for condition identification;

- <origin> is a origin type for this condition;
- <variable> is a variable which should has a condition;
- <comparator> is a comparison operator (EQ, NE, GT, LT, GE, LE, IN, NOTIN);
- <condition> is a condition for variable.

If condition is composite then it is separated to parts and for each of them assigned one common order number. It is necessary each record has all 5 parts of this structure.

Dataset Name	Parameter Identifier	Variable Name	Variable label	Variable Type	Display format	Codelist/Controlled Term	Source/Derivation	Comments	Codelist	Origin
ADVS	Req	USUBJID	Unique Subject Identifier	text	S30		VS.USUBJID			Predecessor
ADVS	Req	PARAM	Parameter	text	S200		Diastolic Blood Pressure (mmHg), Systolic Blood Pressure (mmHg), Pulse Rate (BEATS/MIN), Height (cm). For records with a corresponding record in VS, populate with the value of VS.VSTEST ([VS.VSTRESU]) For records created to contain the 'Body Mass Index-derived' result, populate with PARAM= "Body Mass Index-derived (kg/m2)"		PARAM_ADVS	Derived
ADVS	Req	PARAMCD	Parameter Code	text	S8		DIABP, SYSBP, PULSE, HEIGHT, WEIGHT, BMID @1@VS.VSTESTCD where One-to-one correspondence with PARAM For records with a corresponding record in VS ; @2@If PARAM= "Body Mass Index-derived" result, then PARAMCD= "BMID";		PARAMCD_ADVS	@1@Predecessor@PARAM@NOTIN@'Body Mass Index-derived', 'Heart Rate Recovery'; @2@Derived@PARAM@EQ@'Body Mass Index-derived';
ADVS	Perm	PARAMTYP	Parameter Type	text	S200	DERIVED, null	Populated with "DERIVED" for records created to parameteres 'Heart Rate Recovery' or 'Body Mass Index-derived', blank otherwise			Assigned

In “Source/Derivation” column, for variables with composite origin type, the structure must be:

@<order number>@<full condition description>;

Where <order number> is an index number that corresponds to <order number> in Origin Column and <full condition description> is a text with source/derivation information about each condition.

If in full condition description there is 'then' word, in Derivation column in define.xml file only the part of condition description after this word will be reported. This way the information presentation will be more readable. In the example above on the variable PARAMCD, we can see the two conditions connected to Origin column:

@1@VS.VSTESTCD where One-to-one correspondence with PARAM

For records with a corresponding record in VS ;

@2@If PARAM="Body Mass Index-derived" result, then PARAMCD= "BMID";

AUTOMATIC CREATION OF INPUT FILE FOR PINNACLE 21 COMMUNITY®

Starting with the import of ADaM Metadata file the macro proceed step-by-step in the creation of each sheet that must be included in Excel Spec. A sample of derivation of most significant sheets is shown below:

Dataset	Variable	Label	Origin	Method	Predecessor	Comment
ADVS	USUBJID	Unique Subject Identifier	Predecessor		VS.USUBJID	
ADVS	PARAM	Parameter	Derived	ADVS.PARAM		
ADVS	PARAMCD	Parameter Code				
ADVS	PARAMTYP	Parameter Type	Assigned			ADVS.PARAMTYP

In this example the sheet “Variables” from Excel Spec is shown.

‘Variable Origin’ is equal to ‘Origin’ specified in ADaM Metadata when this is equal to 'Predecessor', 'Derived' or 'Assigned', otherwise it is left blank and the define.xml will get the information on the ‘Origin’ from “ValueLevel” and “WhereClauses” sheets.

Variable ‘Method’ contains the ID of the record the user have to check in sheet “Methods” to find the derivation method of a variable with Origin = 'Derived'. Variable ‘Predecessor’ contains the predecessor variable when Origin = 'Predecessor'.

Variable ‘Comment’ contains the ID of the record the user have to check in sheet “Comments” to find the comment linked to the variable.

Dataset	Variable	Where Clause	Origin	Method	Predecessor	Value Level	Comment	Join Comment
ADVS	PARAMCD	ADVS.PARAM.NOTIN.'BodyMassIndex-derived','HeartRateRecovery'	Predecessor	ADVS.PARAMCD.ADVS.PARAM.NOTIN.'BodyMassIndex-derived','HeartRateRecovery'	VS.VSTESTCD			
ADVS	PARAMCD	ADVS.PARAM.EQ.'BodyMassIndex-derived'	Derived	ADVS.PARAMCD.ADVS.PARAM.EQ.'BodyMassIndex-derived'				

“ValueLevel” sheet above details the composite origin type. ‘Where Clause’ variable is used as ID in “Where Clauses” sheet as shown in the below example.

ID	Dataset	Variable	Comparator	Value
ADVS.PARAM.EQ.'BodyMassIndex-derived'	ADVS	PARAM	EQ	'Body Mass Index-derived'
ADVS.PARAM.NOTIN.'BodyMassIndex-derived','HeartRateRecovery'	ADVS	PARAM	NOTIN	'Body Mass Index-derived','Heart Rate Recovery'

The “Codelist” sheet explains in detail the codelists used. The macro processes the codelist column in ADaM_Metadata comparing the values found in the datasets with CDISC Terminology, notifying the user if some value is misspelled.

To create this sheet the macro reads ADaM Terminology.xls and SDTM Terminology.xls files (previously stored in the same folder of ADaM Metadata) together with Terms and Decoded Values taken from datasets.

ID	Name	NCI Codelist Code	Data Type	Order	Term	NCI Term Code	Decoded Value
ARM	ARM		text	1	Active Treatment		Active Treatment
ARM	ARM		text	2	Not Randomized		Not Randomized
ARM	ARM		text	3	Placebo		Placebo
ARMCD	ARMCD		text	1	A		Active Treatment
ARMCD	ARMCD		text	2	NOTASSGN		Not Randomized
ARMCD	ARMCD		text	3	PBO		Placebo
ARMN	ARMN		text	1		1	Active Treatment
ARMN	ARMN		text	2		2	Placebo
ARMN	ARMN		text	3		3	Not Randomized

Once the Excel Spec is ready this can be processed by Pinnacle 21 Community® through the “Generate Define” facility. The last step the user have to do for completing the process of creation of define.xml is to validate it using the same software. Define.xml should be placed in the same folder with xpt files of datasets to be processed by “Validator” facility, understanding with the result if the Excel Spec file have been created properly.