

Tips and Tricks: From Specification to Define.xml v2.1 using SAS

Caroline Pelckmans
Oussama Zineddine

SI06 – PHUSE EU Connect 2024



AGENDA

- Introduction
- General Process Flow to Generate ADaM Define.xml
- Tips and Tricks to Update Define v2.0 to Define v2.1
- Conclusion



#Data #DataDriven
#DataStrategy #DataIntegration
#DataViz #DataStandards
#DataScience #Datagovernance
#HealthCare #DataAnalysis
#ClinicalTrials
#ArtificialIntelligence
#AdvancedAnalytics
#MachineLearning #Digital
#Chatbot #Biometrics #CRO
#ClinicalData
#LifeSciencesExperts
#Whodrug #MedDRA
#Pharmacovigilance
#ClinicalResearchLifeCycle
#MedicalWriting
#MedicalInformation
#MedicalReview #SDTM
#ADaM #CDISC #Innovation
#RegulatorySubmissions
#Protocol #Coding

INTRODUCTION



INTRODUCTION

Define.xml:

- Crucial machine-readable document
- Outlines the metadata of (submitted) datasets
- Plays a vital role in the regulatory review process
- Version 2.0 published in March 2013 and version 2.1 published in May 2019
- March 2023: the FDA recommends using Define.xml version 2.1 for any study initiated on or after March 15, 2023

Information retrieved from the FDA Data Standards Catalog v10.4					
Use	Property	Date Support Begins	Date Support Ends	Date Requirement Begins	Date Requirement Ends
Study data definition	Define.xmlv1.0	Ongoing	03/15/2018	12/17/2016 [1] 12/17/2017 [2]	03/15/2018
Study data definition	Define.xmlv2.0	08/07/2013		12/17/2016 [1] 12/17/2017 [2]	
Study data definition	Define.xmlv2.1	07/07/2020		03/15/2022 [1] 03/15/2023 [2]	

INTRODUCTION

- Define.xml v2.0 for submitted SDTM, ADaM, or SEND datasets must comprehensively outline the metadata used to describe:
 - (Submitted) datasets
 - Variables
 - Possible values of variables when appropriate
 - Controlled terminologies
 - Codes
- Define.xml v2.1
 - Improved identification of the standards
 - Improved referencing of controlled terminology versions
 - Updated approach to def:Origin

CDISC-Sample

- Supplemental Documents Standards
- Analysis Results Metadata
- Datasets
- Controlled Terminology
- Methods

Expand all VLM

Collapse all VLM

Datasets

Dataset	Description	Class - SubClass	Structure	Purpose	Keys	Documentation	Location
ADSL [ADaMIG 1.1]	Subject-Level Analysis	SUBJECT LEVEL ANALYSIS DATASET	one record per subject	Analysis	STUDYID, USUBJID	Screen Failures are excluded since they are not needed for this study analysis. See referenced dataset creation program and ADRG adsl.sas Analysis Data Reviewer's Guide [6]	adsl.xpt
ADQSADAS [ADaMIG 1.1]	ADAS-Cog Analysis	BASIC DATA STRUCTURE	One record per subject per parameter per analysis visit per analysis date	Analysis	STUDYID, USUBJID, PARAMCD, AVISIT, ADT	See referenced dataset creation program and ADRG adqsadas.sas Analysis Data Reviewer's Guide [Section2.1]	adqsadas.xpt
ADAE [ADaMIG 1.1]	Adverse Events Analysis Dataset	OCCURRENCE DATA STRUCTURE - ADVERSE EVENT	one record per subject per adverse event	Analysis	STUDYID, USUBJID, AETERM, ASTDT, AESEQ	See SAS program adae.sas	adae.xpt

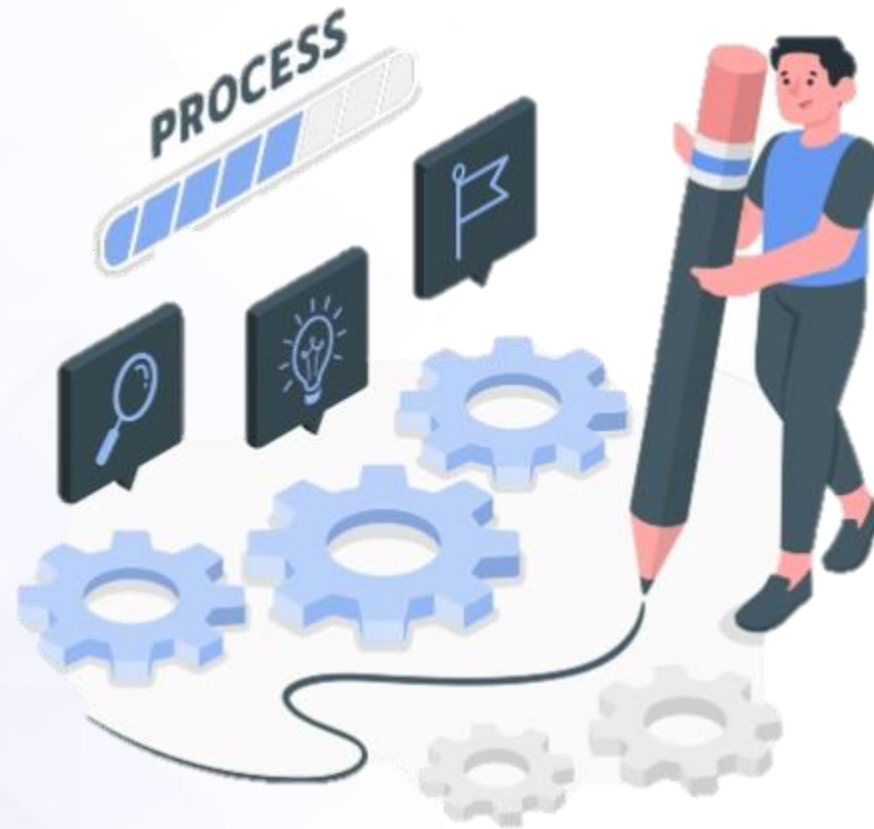
Go to the [top](#) of the Define-XML document

ADSL (Subject-Level Analysis) - [ADaMIG 1.1]

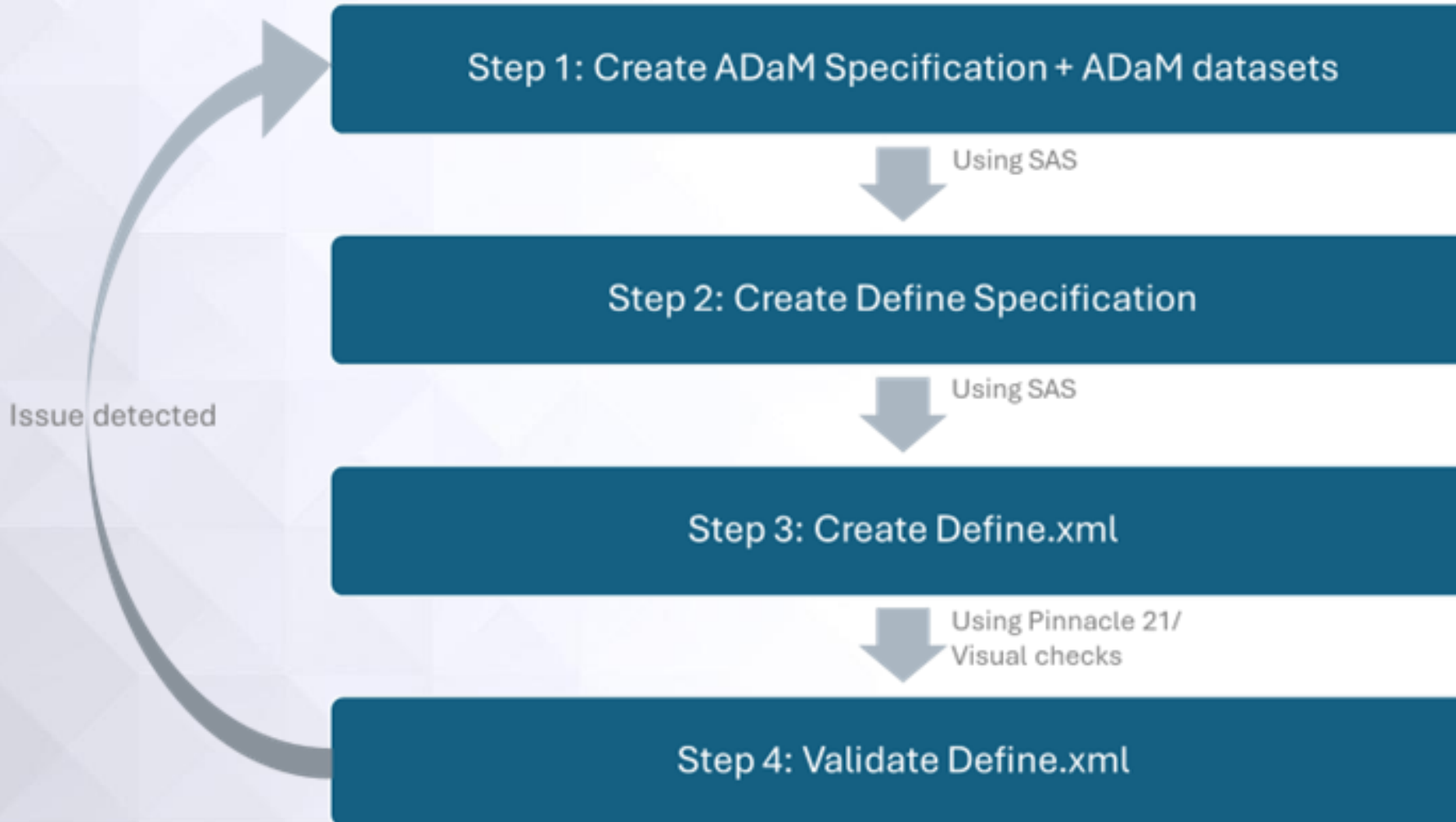
Location: [adsl.xpt](#)

Variable	Label / Description	Type	Length or Display Format	Controlled Terms or ISO Format	Origin / Source / Method / Comment
STUDYID	Study Identifier	text	12		Predecessor: DM.STUDYID
USUBJID	Unique Subject Identifier	text	11		Predecessor: DM.USUBJID

GENERAL PROCESS FLOW



GENERAL PROCESS FLOW



GENERAL PROCESS FLOW

STEP 1: CREATE ADAM SPECIFICATIONS

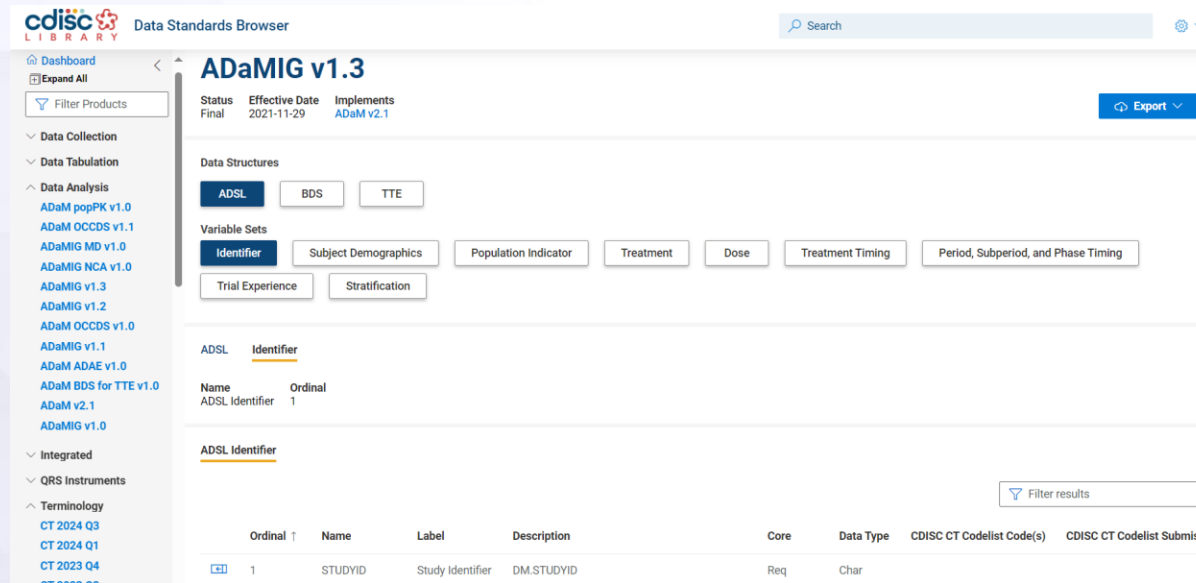
	A	B	C	D	E	F	G	H	I	J
3	DOMAIN PREFIX	VARIABLE NAME	VARIABLE LABEL	VARIABLE TYPE	VARIABLE LENGTH	ROLE	USER DEFINE	CDISC CT	COMP. ALGORITHM	EXTERNAL
4	ADSL	STUDYID	Study Identifier	char	200	Identifier				
5	ADSL	USUBJID	Unique Subject Identifier	char	200	Identifier				
6	ADSL	SUBJID	Subject Identifier for the Study	char	200	Identifier				
7	ADSL	SITEID	Study Site Identifier	char	200	Identifier				
8	ADSL	SITEGRy	Pooled Site Group y	char	200	Identifier			ADSL.SITEGRy	
9	ADSL	SITEGRyN	Pooled Site Group y (N)	num	8	Identifier			ADSL.SITEGRyN	
10	ADSL	REGIONy	Geographic Region y	char	200	Identifier			ADSL.REGIONy	
11	ADSL	REGIONyN	Geographic Region y (N)	num	8	Identifier			ADSL.REGIONyN	

- Our template includes the following sheets:
 - **Table metadata:** Describes the included ADaMs
 - **Variables metadata:** Describes the variables in the ADaMs and their traceability
 - **Value level metadata:** Describes the metadata associated with Basic Data Structure parameters
 - **Codelist metadata:** Describes codelists copied from the SDTM specification or codelists specific to newly created ADaM variables
 - **Computational Algorithms:** Describes the derivations of derived variables
 - **XML parameters:** To describe study-specific parameters such as the study identifier, study description, and supplemental documents

GENERAL PROCESS FLOW

Filling out the ADaM specification template can be time consuming, so here are a few tips to streamline the process:

1. Use the CDISC library (<https://library.cdisc.org/browser/#/>) to your advantage



2. Use conditional formatting in your ADaM Specification
3. Use SAS macros and Excel functions to ease filling in the specification, when possible

GENERAL PROCESS FLOW

STEP 2: TRANSFORM ADAM SPECIFICATION INTO DEFINE SPECIFICATION

The Define specification, similar to the ADaM specification, is an Excel spreadsheet that includes various sheets. Here is a list of the sheets included in our Define specification:

- Instructions
- Table Metadata
- Variables Metadata
- Value Level Metadata
- Computational Algorithm
- Comments
- Standards
- ODM
- Global Variables
- Metadata Version
- Supplemental Docs
- Analysis results Metadata

	A	B	C	D	E	F	G	H	I	J	K	L
1	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
2	Domain	ItemOID	OrderNumber	Mandatory	KeySequence	MethodOID	Role	RoleCodeListOID	OID	Name	DataType	Length
3	IG.ADAE	IT.ADAE.STUDYID	1	No					IT.ADAE.STUDYID	STUDYID	text	10
4	IG.ADAE	IT.ADAE.USUBJID	2	No	1				IT.ADAE.USUBJID	USUBJID	text	18
5	IG.ADAE	IT.ADAE.SUBJID	3	No					IT.ADAE.SUBJID	SUBJID	text	7
6	IG.ADAE	IT.ADAE.SITEID	4	No					IT.ADAE.SITEID	SITEID	text	4
7	IG.ADAE	IT.ADAE.COUNTRY	5	No					IT.ADAE.COUNTRY	COUNTRY	text	3

< > ...

INSTRUCTIONS

TableMetadata

VariablesMetadata

ValueLevelMetadata

CodeListMetadata

ComputationalAlgorithm

Comments

Standards

ODM

(... + : ◀ ▶

GENERAL PROCESS FLOW

STEP 3: GENERATE DEFINE.XML

- Each sheet of the Define specification → SAS dataset
- PROC TEMPLATE codes should be included that define the specific structure for the XML files, using a SAS macro:

```
%MACRO DOINCLUDE(PROGRAMNAME);  
    %INCLUDE "&PROGPATH.\&PROGRAMNAME..sas";  
%MEND DOINCLUDE;
```

- Information from the different Define specification sheets should be combined with the appropriate PROC TEMPLATES to create several XML files:

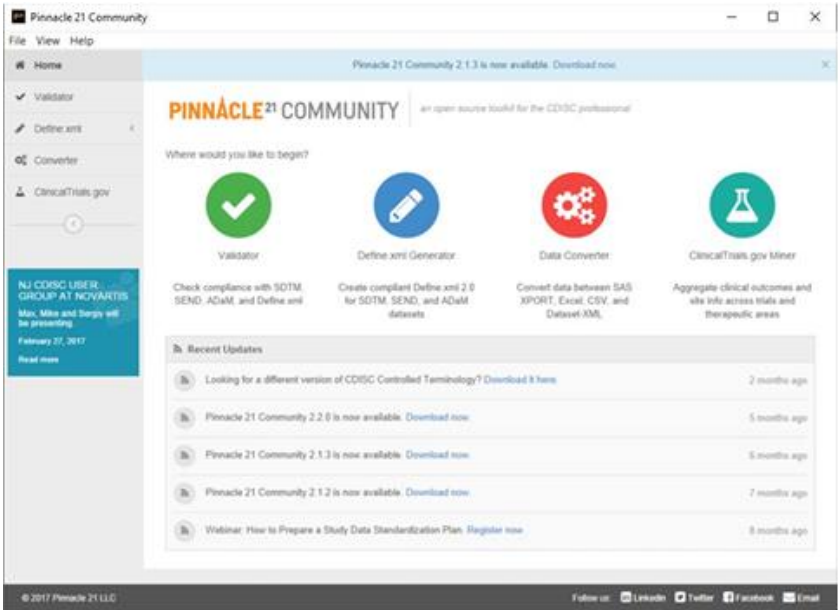
```
LIBNAME TMP_XML XML "&DEFINE.\&STUDY_ID.\Tempdata\FileName.xml!"  
    tagset=TMP.TagsetName;  
PROC COPY IN=WORK OUT=TMP_XML;  
    SELECT SheetName;  
RUN;
```

- The different XML files → the final Define.xml output
- To view the Define.xml in a browser-based report, an XSL stylesheet should be referenced and used.

GENERAL PROCESS FLOW

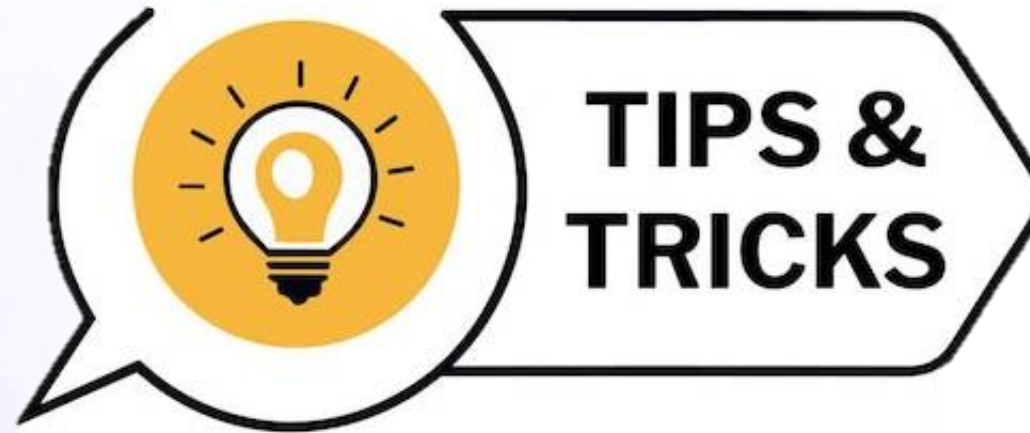
STEP 4: VALIDATE THE CREATED DEFINE.XML

- Creation of Pinnacle 21 report
 - To validate the Define.xml file
 - Run Pinnacle 21 in standalone mode and in combination with the study datasets in v5 Transport Format
- Visual Checks
 - Have another programmer review the Define.xml file
 - QC Checklist



DEFINE MANUAL QC CHECKLIST			Initial of Reviewer:	XXX
		Description		
1	All	Check that the following files are available: .xml, .xsl, compaig.pdf and compaig1000.pdf (if there are algorithms with length>=1000).		
2	All	Check that the standards are correctly mentioned, including ADAMIG.		
3	All	Check that all the needed bookmarks are present in the left part of the define.xml.		
4	All	Check that the study information is correct : study name, description, protocol name, metadata name and description.		
5	All	Check that the layout is correct and the file is easy to read.		
6	All	Check that the external hyperlinks are working by clicking on one of each type and that the target is correct. Note: all the documents need to be in the same file.		
7	All	Check that the internal hyperlinks are working by clicking on one of each type and that you can easily navigate into the file.		
8	All	In the "Analysis Datasets" or "Datasets" part, check that no field is missing.		
9	All	In the "Controlled Terminology" part, check that all the Controlled Terms are defined and have a label.		
10	All	For "VLM", check that Expand and Collapse buttons are working if applicable, check that all necessary items for value level metadata are present (for example, check if all needed PARAMCDs are available under the VLM of AVAL) and are not truncated.		
11	All	In the "Analysis Derivations" or "Methods" part, check that no description is missing or truncated.		
12	All	Confirm Pinnacle21 Report(s) was saved on the sever and that the summary tab has no errors or warnings anymore. If any, verify that it has been justified (by adding a column called "Justification" in the "Issue Summary" tab in the report itself).		

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1



TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

Major enhancements introduced in Define.xml v2.1 that have an impact on the creation of the ADaM Define.xml:

- Identification of CDISC standards using the new elements def:Standards and def:Standard
- Datasets section:
 - Support for Subclasses, using the new element def:SubClass
 - Added reference to applicable ADaM IG for each dataset
- Controlled Terminology referencing the applicable standards
- Updated approach to def:Origin

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

IDENTIFICATION OF CDISC STANDARDS:

In Define.xml v2.1, the def:StandardName and def:StandardVersion attributes were deprecated and replaced by the def:Standards and def:Standard elements

Standards for Study CDISC-Sample			
Standard	Type	Status	Documentation
ADaMIG 1.1	IG	Final	
CDISC/NCI ADaM 2017-09-29	CT	Final	ADaM specific CT is applicable for a few variables only.
CDISC/NCI SDTM 2018-06-29	CT	Final	SDTM CT is applicable for variables copied from SDTM to ADaM.



```
<def:Standards>
  <def:Standard OID="STD.ADaMIG"
    Name="ADaMIG"
    Type="IG"
    Status="Final"
    Version="1.1"/>
  <def:Standard OID="STD.CTA"
    Name="CDISC/NCI"
    Type="CT"
    PublishingSet="ADaM"
    Status="Final"
    Version="2017-09-29"
    def:CommentOID="COM.STDCT.01"/>
  <def:Standard OID="STD.CTS"
    Name="CDISC/NCI"
    Type="CT"
    PublishingSet="SDTM"
    Status="Final"
    Version="2018-06-29"
    def:CommentOID="COM.STDCT.02"/>
</def:Standards>
<def:CommentDef OID="COM.STDCT.01">
  <Description>
    <TranslatedText xml:lang="en">ADaM CT is applicable for a few variables
    only.
    </TranslatedText>
  </Description>
</def:CommentDef>
<def:CommentDef OID="COM.STDCT.02">
  <Description>
    <TranslatedText xml:lang="en">SDTM CT is applicable for variables
    copied from SDTM to ADaM.
    </TranslatedText>
  </Description>
</def:CommentDef>
```

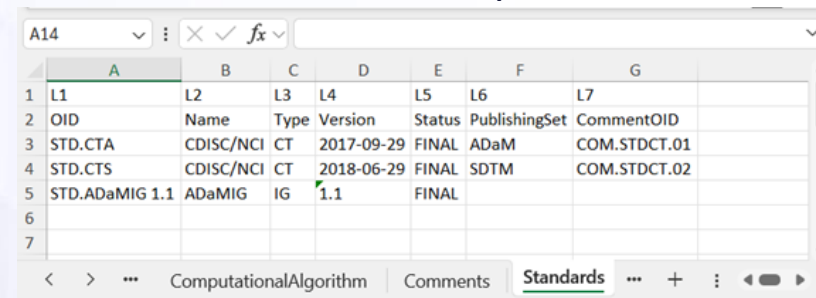
TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

To create it, we recommend to adjust only Steps 2 and 3 of the general process flow.

- **For Step 2:** Use macro variables:

```
Data Standards;  
  LENGTH L1-L7 $200.;  
  RETAIN L1-L7;  
  L1="STD.CTA"; L2="CDISC/NCI"; L3="CT"; L4="&ADAMCT."; L5="FINAL"; L6="ADAM";  
  L7="COM.STDCT.01"; output;  
  L1="STD.CTS"; L4="&SDTMCT."; L6="SDTM"; L7="COM.STDCT.02"; output;  
  L1="STD.ADaMIG"; L2="ADaMIG"; L3="IG"; L4="&ADAMIG."; L6=""; L7=""; output;  
  
Run;
```

The information gathered via the above SAS code will be presented in the Define specification as follows:



	A	B	C	D	E	F	G
1	L1	L2	L3	L4	L5	L6	L7
2	OID	Name	Type	Version	Status	PublishingSet	CommentOID
3	STD.CTA	CDISC/NCI	CT	2017-09-29	FINAL	ADaM	COM.STDCT.01
4	STD.CTS	CDISC/NCI	CT	2018-06-29	FINAL	SDTM	COM.STDCT.02
5	STD.ADaMIG 1.1	ADaMIG	IG	1.1	FINAL		
6							
7							

- **For Step 3:** The information stored in the Define specification can be used to create the Define.xml using SAS:

```
%DOINCLUDE (Standards);  
LIBNAME TMP_XML XML "&DEFINE.\&STUDY_ID.\Tempdata\STD.xml!" tagset=TMP.standards;  
PROC COPY IN=WORK OUT=TMP_XML;  
  SELECT STANDARDS;  
  
RUN;
```

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

DATASETS SECTION:

- **Support for Subclasses** - In Define.xml v2.1 a new child element of def:Class is introduced: def:Subclass, to further group or classify the datasets, when applicable.
- **Reference to applicable ADaM IG for each dataset** – In Define.xml v2.1 dataset metadata is also referencing the applicable ADaMIG version for each dataset separately.

Datasets							
Dataset	Description	Class - SubClass	Structure	Purpose	Keys	Documentation	Location
ADSL [ADaMIG 1.1]	Subject-Level Analysis Dataset	SUBJECT LEVEL ANALYSIS DATASET	One record per subject	Analysis	USUBJID	See ADRG Section 5.2.1 ADSL for more details. Reviewer's Guide	adsl.xpt
ADAE [ADaMIG 1.1]	Adverse Events Analysis Dataset	OCCURRENCE DATA STRUCTURE • ADVERSE EVENT	one record per subject per adverse event	Analysis	USUBJID, AETERM, ASTDT, AESEQ		adae.xpt



```
<ItemGroupDef OID="IG.ADAE"
  Name="ADAE"
  Repeating="Yes"
  IsReferenceData="No"
  SASDatasetName="ADAE"
  Purpose="Analysis"
  def:Structure="one record per subject per adverse event"
  def:StandardOID="STD.ADaMIG 1.3"
  def:ArchiveLocationID="LF.ADAE">
...
  <def:Class Name="OCCURRENCE DATA STRUCTURE">
    <def:SubClass Name="ADVERSE EVENT"/>
  </def:Class>
...
</ItemGroupDef>
```

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

For Step 1: We recommend updating the ADaM specification template.

	A	B	C	D	E	F	G	H	I
1	B1	B2	B3	B4	B5	B7	B9	B10	B11
2	Dataset	Description	Structure	Purpo	Keys	Class	Documentati	Standard	Subclass
3	ADSL	Subject-Level Analysis Dataset	One record per subject	Analysis	USUBJID	ADSL	See ADRG Section	ADaMIG 1.3	
4	ADAE	Adverse Events Analysis Dataset	one record per subject per adverse event	Analysis	USUBJID, AETERM, ASTDT, AESEQ	OCCDS		ADaMIG 1.3	ADVERSE EVENT
5	ADTTE	Time to Event Analysis Dataset	One record per subject per parameter	Analysis	USUBJID, PARAMCD	BDS	See ADRG Section	ADaMIG 1.3	TIME-TO-EVENT

To avoid any Pinnacle 21 issues, we recommend to provide a drop-down list for Subclass.

For Step 2: SAS code used to create the Define specification needs to be updated to also get the Subclass information into the Define specification under the TableMetadata sheet:

	A	B	C	D	G	H	I	J	K	L	P	Q
1	A1	A2	A3	A4	A7	A8	A9	A10	A11	A12	A16	A17
2	OID	Name	Domain	SASDatasetName	Purpose	Structure	Class	def:Archive LocationID	def:CommentOID	Label	StandardOID	SubClass
3	IG.ADSL	ADSL	ADSL	ADSL	Analysis	One record per subject	SUBJECT LEVEL ANALYSIS DATASET	LF.ADSL	COM.ADSL	Subject-Level Analysis Dataset	STD.ADaMIG 1.3	
4	IG.ADAE	ADAE	ADAE	ADAE	Analysis	one record per subject per adverse event	OCCURRENCE DATA STRUCTURE	LF.ADAE		Adverse Events Analysis Dataset	STD.ADaMIG 1.3	ADVERSE EVENT
5	IG.ADTTE	ADTTE	ADTTE	ADTTE	Analysis	One record per subject per parameter	BASIC DATA STRUCTURE	LF.ADTTE	COM.ADTTE	Time to Event Analysis Dataset	STD.ADaMIG 1.3	TIME-TO-EVENT

For Step 3: The information stored in the Define specification, will need to be used to create the Define.xml

```
%DOINCLUDE (ItemGroupDef);
LIBNAME TMP_XML XML "&DEFINE.\&STUDY_ID.\Tempdata\ItemGroupDef.xml" tagset=TMP.ItemGroupDef;
PROC COPY IN=WORK OUT=TMP_XML;
    SELECT ItemGroupDef;
RUN;
```

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

CONTROLLED TERMINOLOGY REFERENCING APPLICABLE STANDARDS:

- The controlled terminology section in Define v2.1 is enhanced by referencing the CT standard version. When the controlled terminology is sponsor defined, a `def:IsNonStandard` attribute is used to indicate that the codelist is "Non Standard":
- XML file will need to include `def:IsNonStandard` and `def:StandardOID` (when applicable):

Subcategory for Question [Non Standard]

Permitted Value (Code)
ADDITIONAL CONCERNS
EMOTIONAL WELL-BEING
FUNCTIONAL WELL-BEING
PHYSICAL WELL-BEING
SOCIAL/FAMILY WELL-BEING

Race [C74457] [CDISC/NCI SDTM 2020-12-18]

Permitted Value (Code)
ASIAN [C41260]
BLACK OR AFRICAN AMERICAN [C16352]
MULTIPLE [*]
NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER [C41219]
OTHER [*]
WHITE [C41261]

* Extended Value

```
</CodeList>
<CodeList OID="CL.QSSCAT"
  Name="Subcategory for Question"
  DataType="text"
  def:IsNonStandard="Yes">
...
</CodeList>

</CodeList>
<CodeList OID="CL.RACE"
  Name="Race"
  DataType="text"
  def:StandardOID="STD.CTS"
  SASFormatName="$RACE">
<EnumeratedItem CodedValue="ASIAN"
  OrderNumber="1">
  <Alias Name="C41260" Context="nci:ExtCodeID"/>
</EnumeratedItem>
...
</CodeList>
```

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

To implement it, we recommend to adjust SAS codes for Steps 2 and 3 of the general process flow:

- **For step 2:**

- Add 2 additional columns to the CodeListMetadata sheet :

- ✓ **StandardOID**
- ✓ **IsNonStandard**

A	B	C	D	E	F	G	I	M	O	X	Y
D1	D2	D3	D4	D5	D6	D7	D9	D13	D15	D24	D25
OID	Name	DataType	SASFormatName	AliasContext	AliasName	CodedValueEnum	OrderNumberEnum	CodedValueCode	OrderNumberCode	StandardOID	IsNonStandard
CL.QSSCAT	Subcategory for Question	text				ADDITIONAL CONCERNS	1				Yes
CL.QSSCAT	Subcategory for Question	text				EMOTIONAL WELL-BEING	2				Yes
CL.QSSCAT	Subcategory for Question	text				FUNCTIONAL WELL-BEING	3				Yes
CL.QSSCAT	Subcategory for Question	text				PHYSICAL WELL-BEING	4				Yes
CL.QSSCAT	Subcategory for Question	text				SOCIAL/FAMILY WELL-BEING	5				Yes
CL.RACE	Race	text	\$RACE	nci:ExtCodeID	C74457	ASIAN	1			STD.CTS	
CL.RACE	Race	text	\$RACE	nci:ExtCodeID	C74457	BLACK OR AFRICAN AMERICAN	2			STD.CTS	
CL.RACE	Race	text	\$RACE	nci:ExtCodeID	C74457	MULTIPLE	3			STD.CTS	
CL.RACE	Race	text	\$RACE	nci:ExtCodeID	C74457	NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER	4			STD.CTS	
CL.RACE	Race	text	\$RACE	nci:ExtCodeID	C74457	OTHER	5			STD.CTS	
CL.RACE	Race	text	\$RACE	nci:ExtCodeID	C74457	WHITE	6			STD.CTS	

- **For step 3:**

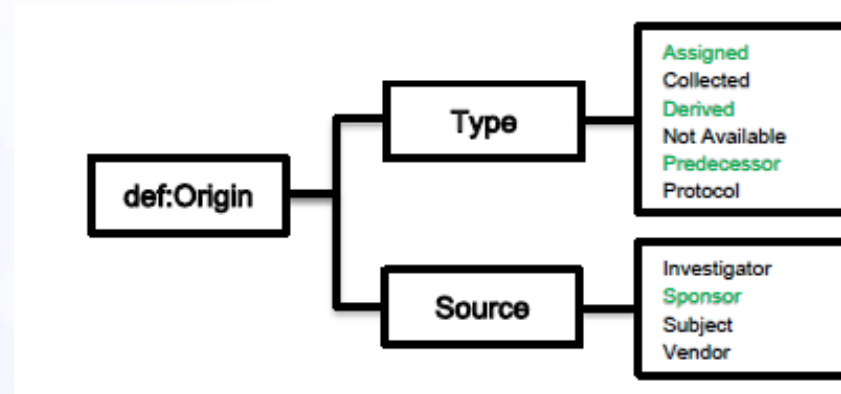
- Use the following SAS Code:

```
%DOINCLUDE (CodeList);  
LIBNAME TMP_XML XML "&DEFINE.\&STUDY_ID.\Tempdata\CodeList.xml" tagset=TMP.CodeList;  
PROC COPY IN=WORK OUT=TMP_XML;  
    SELECT CodeList;  
RUN;
```

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

ORIGIN ENHANCEMENT:

- The representation of origin metadata was enhanced to identify the source in addition to the origin details:



- The possible values for the Type attribute remain unchanged between ADaM Define v2.0 and v2.1:
 - Assigned
 - Derived
 - Predecessor
- For the Source attribute, only the value 'Sponsor' is used in the ADaM Define for Derived and Assigned origin Types.

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

TRTEMFL	Treatment Emergent Analysis Flag	text	1	<u>No Yes Response - Yes Only</u> "Y" = "Yes"	Derived (Source: Sponsor) Equals "Y" when not missing ASTDT and not missing TRTSDT and TRTSDT <= ASTDT <= ENDT Else set to null. Note: ENDT equal to SV.SVSTDTC when SV.VISITNUM=11 and ADSL.DISCTRFL is missing ;ENDT equal to ADSL.DISCTRDT + 14 when ADSL.DISCTRFL = "Y"
---------	----------------------------------	------	---	---	---



```
</ItemDef>
<ItemDef OID="IT.ADAE.TRTEMFL"
  Name="TRTEMFL"
  DataType="text"
  Length="1"
  SASFieldName="TRTEMFL">
  <Description>
    <TranslatedText xml:lang="en">Treatment Emergent Analysis Flag</TranslatedText>
  </Description>
  <CodeListRef CodeListOID="CL.Y BLANK"/>
  <def:Origin Type="Derived"
    Source="Sponsor">
  </def:Origin>
</ItemDef>
```

TIPS AND TRICKS TO UPDATE DEFINE V2.0 TO DEFINE V2.1

To include it, we recommend to only adjust Steps 2 and 3 of the general process flow:

- **For Step 2:** Add a simple step in SAS code to assign the Source to Sponsor when a variable is Derived or Assigned:

```
if upcase(col20) in ('DERIVED','ASSIGNED') and COL28='' then COL28='Sponsor';
```

	A	B	C	I	J	K	L	Q	S	T	AB
1	B1	B2	B3	B9	B10	B11	B12	B17	B19	B20	B28
2	Domain	ItemOID	OrderNumber	OID	Name	DataType	Length	CodeListOID	Description	Origin	Source
71	IG.ADAE	IT.ADAE.TRTEMFL	69	IT.ADAE.TRTEMFL	TRTEMFL	text	1	CL.Y_BLANK	Treatment Emergent Analysis Flag	Derived	Sponsor
1239											
	VariablesMetadata ValueLevelMetadata CodeListMetadata ComputationalAlgorithm Comments Standards ODM GlobalVariables Metadata +										

- **For Step 3:**

- Use SAS Code

```
%DOINCLUDE (ItemDef);  
LIBNAME TMP_XML XML "&DEFINE.\&STUDY_ID.\Tempdata\ItemDef.xml" tagset=TMP.ItemDef;  
PROC COPY IN=WORK OUT=TMP_XML;  
    SELECT ItemDef;  
RUN;
```

- PROC COPY can be used to include the information from the Define specification, which is stored in a SAS dataset named ItemDef, to create the corresponding ItemDef.XML.

CONCLUSION



CONCLUSION



- As the FDA recommends using Define.xml version 2.1 for any study initiated on or after March 15, 2023, it is of interest that your programming team can also produce this version.
- Using the tips and tricks given in this presentation should ease the upgrade to create Define v2.1 when using an internal SAS tool that goes from an ADaM specification to a Define specification to a Define.xml.



THANK YOU FOR YOUR ATTENTION **Q&A**

CONTACT



Caroline PELCKMANS

Biostatistician /

Project Manager

caroline.pelckmans@businessdecision.be



Oussama ZINEDDINE

Senior Statistical Programmer

oussama.zineddine@businessdecision.be