PharmaSUG 2022 - Paper SS-023

Lessons Learned from Using P21 to Create ADaM define.xml with Examples

Yizhuo Zhong, Christine Teng, Majdoub Haloui, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

Define.xml is a required component for the FDA, PMDA and NMPA submission data packages. It provides metadata for collected data as well as derived analysis data. Pinnacle 21 Enterprise (P21E) is a web-based tool that is commonly used by many sponsors and CROs to generate the define.xml. P21E supports multiple versions of CDISC standards and regulatory agency's business rules. During validation of ADaM datasets and ADaM define.xml, error and warning messages will be reported by P21E based on the configuration of data package used. These issues must be addressed to comply with CDISC and agency business rules. This paper will focus on the lessons learned through some examples when creating ADaM define.xml using P21E. A flow chart of suggested process and checking steps will be illustrated. Some best practices on ADaM datasets specification development will be discussed as well.

INTRODUCTION

Define.xml is an essential part of an ADaM submission package. It describes the structure and contents of CDISC ADaM datasets submitted to regulatory agencies. P21E has many useful features, including validating ADaM and define.xml, as well as generating define.xml version 2.0. During ADaM define.xml creation using P21E, many errors or warnings can be identified, impacting the consistency and accuracy of the define.xml file. Such issues need to be identified and diagnosed correctly. Although addressing these issues can be tedious and time-consuming, this is necessary for the quality of define.xml, and much of the checking can be automated by writing utility macros.

This paper will provide an overview of ADaM define.xml creation workflow and will discuss the lessons learned when creating define.xml using P21E, with examples provided. The following issues will be covered in this paper:

- Variable origins for variable/value level metadata
- Length attributes for numeric and date/time variables
- Significant digits for float data type variables
- Codelist terms, including extensible and non-extensible codelists
- Traceability checking using SDTM datasets

ADAM DEFINE.XML GENERATION PROCESS FLOW

P21E itself has a module that creates define.xml. However, this module has some limitations when creating value level metadata for variables other than AVAL and AVALC. Therefore, we have collaborated with the P21E team to create a customized define.xml generator (Merck ADaM define Adaptor) to help generate the define.xml based on our ADaM dataset specifications in Excel format (Display 1.1). This customized software allows statistical programmers to focus more on the metadata content of the study, rather than the complex define.xml syntax.

Display 1.1 ADaM define.xml Creation Using Customized Define Adaptor

To create ADaM define.xml using P21E, a company standard specification template is suggested to be used and updated with study-specific changes. After loading the ADaM dataset specifications and the define.xml is generated, the P21E template specification file needs to be exported. The exported specification file will be updated with the length issues for DTC variable fixed, and the significant digits populated. Then the updated P21E specifications will be imported again with the datasets. Any P21E issues in datasets or specifications identified need to be fixed or documented in Analysis Data Reviewer's Guide (ADRG). After all the issues are addressed, the final define.xml can be exported. Figure 1 below shows the flow chart of this process.

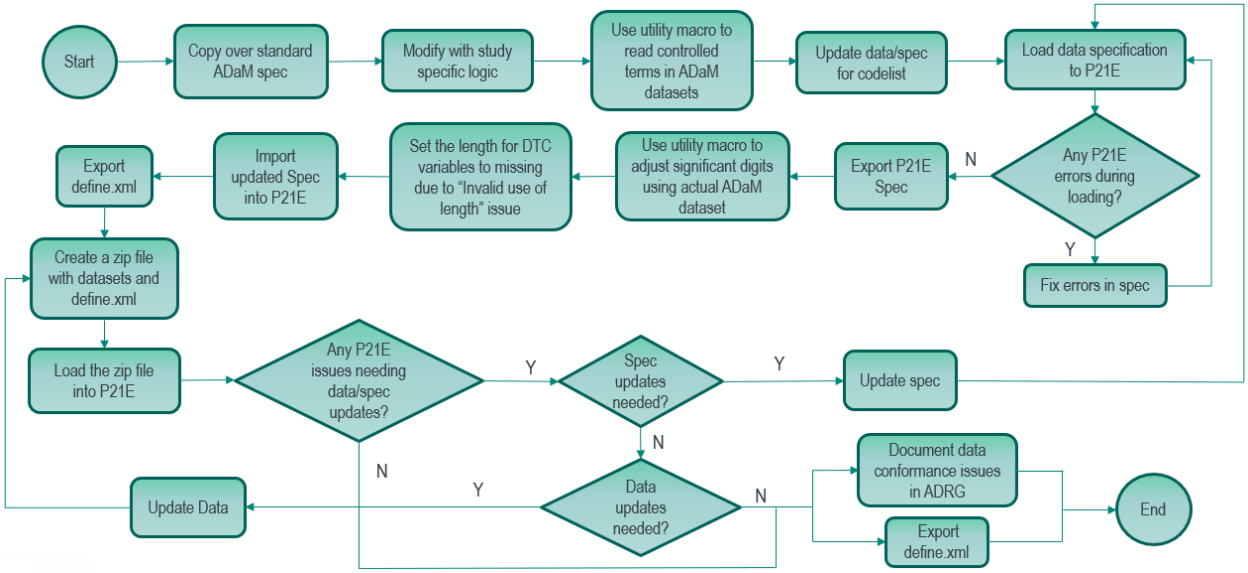

Figure 1 ADaM Define.xml Creation Workflow

COMMON P21E ISSUES DURING DEFINE.XML CREATION

While following the workflow above, some additional points need to be considered to create an ADaM define.xml with better descriptions of attributes, origins, and controlled terms for specific variables. This section will discuss some common issues identified during define.xml creation with suggested solutions.

1. VARIABLE/VALUE LEVEL ORIGIN ISSUES – RULE ID DD0074

There is a common issue related to the origin between variable and value level metadata with Rule ID DD0074, showing that "Variable/Value Level Origin Type mismatch" (Display 2.1).

Variables	Values	Rule ID	Publisher ID	Message	Category	FDA	PMDA
Dataset Name, Variable Name, ValueLevel Variable Where Clause, Variable Origin, ValueLevel Variable Origin	ADINTDT, ADT, ADINTDT.PARAMCD.EQ.IR CPDDT, Assigned, Derived	DD0074		Variable/Value Level Origin Type mismatch	Terminology	Error	Error
Dataset Name, Variable Name, ValueLevel Variable Where Clause, Variable Origin, ValueLevel Variable Origin	ADINTDT, ADT, ADINTDT.PARAMCD.EQ.IN VPDDT, Assigned, Derived	DD0074		Variable/Value Level Origin Type mismatch	Terminology	Error	Error

Display 2.1 P21E ADaM Validation Report with Rule ID DD0074 Issue

For a variable with value level entries, if the origin is specified at variable level, P21E will assume that all the value level entries have the same origin as that in variable level. However, there are cases that a derived variable in ADaM datasets can have different origins in value level metadata.

To work around this issue, the origin can be omitted at variable level when there are multiple value level entries with different origins. Display 2.2 and 2.3 below show an example of the definition of ADT at variable and value level, respectively. The origin of ADT is set to missing at variable level (Display 2.2), since there are origins as “Assigned” and “Derived” among value level entries (Display 2.3).

Ord	Dataset	Variabl	Label	Data Typ	Leng	Significant Digits	Format	Mandat	Codelist	Origin
26	ADINTDT	ADT	Analysis Date	integer	8		yymmdd1	Yes		

Display 2.2 Variable Metadata for ADT in P21E Specifications

Ord	Dataset	Variabl	Where Clause	Descriptio	Data Typ	Leng	Significant Digits	Form	Mandat	Codelis	Origin
1	ADINTDT	ADT	ADINTDT.PARAMCD.EQ.235dc7e627dae945f86		integer	8			No		Assigned
4	ADINTDT	ADT	ADINTDT.PARAMCD.EQ.75da675afa72936d5b4		integer	8			No		Assigned
7	ADINTDT	ADT	ADINTDT.PARAMCD.EQ.27471e0711c6d32cdca		integer	8			No		Derived

Display 2.3 Value Level Metadata for ADT in P21E Specifications

2. VARIABLE LENGTH ISSUES

2.1 Length Issues with Numeric Variables – Rule ID SD1231

For the length of a variable, the error messages with Rule ID SD1231 can be identified by P21E. This shows that the value(s) for a variable is(are) longer than the length defined in the specifications. See Display 3.1 below using AVAL as an example.

Values	Rule ID	Publisher ID	Message	Category	FDA	PMDA
AVGDD, 6147.69230769	SD1231		AVAL value is longer than defined max length 8 when PARAMCD == 'AVGDD'	Limit	Error	Warning
AVGDD, 5532.92307692	SD1231		AVAL value is longer than defined max length 8 when PARAMCD == 'AVGDD'	Limit	Error	Warning

Display 3.1 P21E Validation Report with Rule ID SD1231 Issue

The length defined in define.xml is different from that specified in SAS datasets. In define.xml, the length of a variable refers to the maximum length of the values expressed in characters. Therefore, the length of a numeric (including integer and float) variable can be exceeding 8 characters.

To resolve this issue, the actual length needs to be updated in both variable and value level metadata of the data specifications when it exceeds 8. Display 3.2 shows the original AVAL definition, and Display 3.3 provides the definition of AVAL in the updated specifications. The length of AVAL has been updated to 20 to cover the length of actual values. Note that significant digits of AVAL are also updated, which will be discussed later.

Dataset	Variable	Label	Data Typ	Leng	Significant Digits
ADEXSUM	AVAL	Analysis Value	float	8	

Display 3.2 Original AVAL Definition in P21E Specifications

Dataset	Variable	Label	Data Typ	Leng	Significant Digit
ADEXSUM	AVAL	Analysis Value	float	20	10

Display 3.3 Updated AVAL Definition in P21E Specifications

2.2 Length Issue with Date/Time Variables – Rule ID DD0068

Another common issue with variable length is the one with Rule ID DD0068 showing "Invalid use of Length" (Display 3.4).

Values	Rule ID	Publisher ID	Message	Category	FDA	PMDA
ADCM, CMSTDTC, 19, datetime, ADCM.CMSTDTC	DD0068		Invalid use of Length	Presence	Error	Error
ADCM, CMENDTC, 19, datetime, ADCM.CMENDTC	DD0068		Invalid use of Length	Presence	Error	Error
ADINTDT, SRCDTC, 19, datetime, ADINTDT.SRCDTC	DD0068		Invalid use of Length	Presence	Error	Error

Display 3.4 P21E Validation Report with Rule ID DD0068 Issue

This error is related to the length of "datetime" variables (DTC variables) with "Predecessor" origin. These variables are carried from SDTM datasets "as-is", with the same value, same data type, and label. The DTC variables impacted with this length error include but are not limited to these listed in Table 3.1.

Dataset	Variable	Label
ADAE	AESTDTC	Start Date/Time of Adverse Event
ADAE	AEENDTC	End Date/Time of Adverse Event
ADCM	CMSTDTC	Start Date/Time of Medication
ADCM	CMENDTC	End Date/Time of Medication
ADEX	EXSTDTC	Start Date/Time of Treatment
ADEX	EXENDTC	End Date/Time of Treatment
ADLB	LBDTC	Date/Time of Specimen Collection
ADSL	RFSTDTC	Subject Reference Start Date/Time
ADSL	RFENDTC	Subject Reference End Date/Time

Table 3.1 DTC Variables of Predecessor Origin

According to P21E validation rules, the length attributes must be missing when the data type is not "Integer", "Float" or "Text" when creating define.xml. Therefore, to resolve this issue, the length values for DTC variables need to be manually updated to missing in P21E specifications. Display 3.5 shows the original the definition of variable RFSTDTC in define.xml with length shown as 19. Display 3.6 provides the updated definition of RFSTDTC, with length set to missing.

Dataset	Variable	Label	Data Typ	Leng
ADSL	RFSTDTC	Subject Reference Start Date/Time	datetime	19

Display 3.5 Original RFSTDTC Definition in P21E Specifications

Dataset	Variable	Label	Data Typ	Leng
ADSL	RFSTDTC	Subject Reference Start Date/Time	datetime	

Display 3.6 Updated RFSTDTC Definition in P21E Specifications

3. SIGNIFICANT DIGIT ISSUES – RULE ID OD0071

Regarding significant digits of float variables, the issue with Rule ID OD0071 showing "Missing Significant Digits value" is commonly identified (Display 4.1).

Values	Rule ID	Publisher ID	Message	Category	FDA	PMDA
ADTL, AVAL, ADTL.PARAMCD.EQ.STLDIAM, float	OD0071		Missing Significant Digits value	Presence	Error	Warning
ADTL, AVAL, ADTL.PARAMCD.EQ.NEWLS, float	OD0071		Missing Significant Digits value	Presence	Error	Warning

Display 4.1 P21E Validation Report with Rule ID OD0071 Issue

Significant digits are the number of digits following the decimal point in a float number, which is required for float variables. To resolve this issue, it is suggested to create utility macros to update significant digits in P21E specifications and import again with the changes. Display 4.2 provides an example about the updated P21E specifications, with significant digits populated for float variables.

Dataset	Variable	Label	Data Typ	Leng	Significant Digit
ADAE	ADURN	Analysis Duration (N)	float	8	4
ADAE	AEDURDD	Duration of Adverse Event Diff of Dates	float	8	4
ADEX	EXNUMDOS	Number of Daily Doses	float	8	2

Display 4.2 Sample Updated P21E Specifications with Significant Digits Populated

4. ISSUES WITH CODELISTS

4.1 Codelist Consistency Issues – Rule ID SD0037

A common issue during codelist checking is the error with Rule ID SD0037, which is related to codelist terms. This issue shows that some value(s) exist(s) in the data but not in define.xml (Display 5.1).

Values	Rule ID	Publisher ID	Message	Category	FDA	PMDA
Duration on Therapy (days)	SD0037		Value for PARAM not found in (ADEXSUM0PARAM) user-defined codelist	Terminology	Error	Error
Duration on Therapy (days)for Study Medication	SD0037		Value for PARAM not found in (ADEXSUM0PARAM) user-defined codelist	Terminology	Error	Error

Display 5.1 P21E Validation Report with Rule ID SD0037 Issue

P21E will cross-check the codelist terms defined in the specifications and the corresponding values in the datasets. The issue with Rule ID SD0037 will show up when any inconsistencies are identified. To avoid this issue, the values in the actual data should be present in the terms listed in the "Codelist" tab in the specifications. Utility macros can be developed to help check the codelist terms between the data and specifications.

4.2 Extensible and Non-Extensible Codelist Issues – Rule ID CT2001 and CT2002

For extensible and non-extensible codelists, the issues with Rule IDs CT2001 and CT2002 are often encountered in P21E validation report (Display 5.2).

Values	Rule ID	Publisher ID	Message	Category	FDA	PMDA
UN	CT2001		SEX value not found in 'Sex' non-extensible codelist	Terminology	Error	Reject
MULTIPLE	CT2002		RACE value not found in 'Race' extensible codelist	Terminology	Warning	Warning

Display 5.2 P21E Issues for Extensible and Non-Extensible Codelists

During P21E checking, codelist terms will be cross-checked with CDISC Controlled Terminology (CT) file. Any values that are not among those in the CT file will result in issues with Rule ID CT2001 or CT2002.

Display 5.3 and 5.4 below provide more information related to the Rule ID CT2001 issue with the SEX variable. According to the CDISC CT file, the codelist for SEX is non-extensible. The only valid values are "M", "F", "UNDIFFERENTIATED" and "U" with associated National Cancer Institute (NCI) codes (Display 5.3).

SEX (Sex)

NCI Code: C66731, Codelist extensible: No

C66731	SEX			
NCI Code	**CDISC Submission Value**	**CDISC Synonym**	**CDISC Definition**	**NCI Preferred Term**
C16576	F	Female	A person who belongs to the sex that normally produces ova. The term is used to indicate biological sex distinctions, or cultural gender role distinctions, or both. (NCI)	Female
C20197	M	Male	A person who belongs to the sex that normally produces sperm. The term is used to indicate biological sex distinctions, cultural gender role distinctions, or both. (NCI)	Male
C17998	U	U;UNK;Unknown	Not known, not observed, not recorded, or refused. (NCI)	Unknown
C45908	UNDIFFERENTIATED		A person (one of unisexual specimens) who is born with genitalia and/or secondary sexual characteristics of indeterminate sex, or which combine features of both sexes. (NCI)	Intersex

Display 5.3 SEX Codelist in CDISC Controlled Terminology

Suppose that in the actual data, “UN” instead of “UNDIFFERENTATIED” is used in SEX values. Then when displayed in define.xml, there is no associate NCI code for “UN”, since it is not a valid value according to the codelist. Instead, it will show a star (*), indicating that this is an extended value (Display 5.4).

SEX [*C66731*]

Permitted Value (Code)
M [*C20197*]
F [*C16576*]
UN [*]
U [*C17998*]

* Extended Value

Display 5.4 SEX Codelist in the Define.xml of Actual Datasets

5. TRACEABILITY CHECKING ISSUES

SDTM datasets AE, DM and EX should be included with ADaM datasets (for individual studies, not for ISS and ISE) to avoid issues with Rule IDs AD1024, AD1025, AD1026 (Display 6.1). P21E will perform traceability checking using these 3 SDTM datasets as the source.

Rule ID	Publisher ID	Message	Category	FDA	PMDA
AD1024		Traceability rules not executed due to missing DM dataset	Presence	Error	NA
AD1025		Traceability rules not executed due to missing AE dataset	Presence	Error	NA
AD1026		Traceability rules not executed due to missing EX dataset	Presence	Error	NA

Display 6.1 P21E ADaM Validation Report with Traceability Issues

Display 6.2 below shows an example of the error when checking the traceability of ADAE records to SDTM.AE. This message can help identify potential data or programming issues.

Dataset	Record	Count	Rule ID	Publisher ID	Message	Category	FDA	PMDA
ADAE	4791		AD0258		Record key from ADaM ADAE is not traceable to SDTM.AE (extra ADAE recs)	Cross-reference	Error	Error

Display 6.2 P21E ADaM Validation Report with ADAE Traceability Issues

BEST PRACTICES ON DATA SPECIFICATION DEVELOPMENT

Based on the discussions above, the ADaM data specifications serves as a key input for the define.xml, so the information in the specifications must be specific, accurate and consistent. Therefore, some best practices on ADaM specification development are proposed here.

- Use plain English in define derivations and avoid using SAS functions and codes. Ensure no grammatical errors.
- Do not use ambiguous abbreviations.
- Do not use terms/statements which are irrelevant to the study, datasets, and variables.
- Use uppercase letters for variable names throughout the specification file.
- Use consistent wordings throughout the specification file.
- Remove special characters to avoid display issues in define.xml.
- The values described in the specifications must be consistent with the actual values in the datasets.
- For codelists, the codelist ID should be consistent with the corresponding ID in “Codelist” tab.
- For value level metadata, different origins and/or derivation algorithms among all value level entries must be clearly specified.

CONCLUSION

To create a compliant and complete ADaM define.xml, P21E has been broadly used due to proven benefits and easy solutions. The lessons learned during define.xml creation can help us resolve the issues during P21E validation and prevent the issues from happening during the development of ADaM specifications and creation of define.xml. The proposed solutions to the issues, as well as the best practices, are recommended to help enhance the precision of ADaM dataset and variable descriptions in define.xml for regulatory submission.

REFERENCES

1. ADaM Implementation Guide (ADaM-IG v1.1) https://www.cdisc.org/system/files/members/standard/foundational/adam/ADaMIG_v1.1.pdf
2. CDISC Controlled Terminology https://datascience.cancer.gov/resources/cancer-vocabulary/cdisc-terminology
3. Pinnacle 21 ADaM Validation Rules https://www.pinnacle21.com/validation-rules/adam
4. Majdoub Haloui, Hong Qi. “Supplementary Steps to Create a More Precise ADaM define.xml in Pinnacle 21 Enterprise”. PharmaSUG 2020. https://www.lexjansen.com/pharmasug/2020/SS/PharmaSUG-2020-SS-151.pdf
5. Sergiy Sirichenko, Max Kanevsky. Diagnostics of Technical Errors in define.xml File. PharmaSUG 2018. https://www.pharmasug.org/proceedings/2018/SS/PharmaSUG-2018-SS14.pdf

ACKNOWLEDGEMENTS

The authors would like to give special thanks to the management for their review and inputs on the paper.

CONTACT INFORMATION

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

Yizhuo Zhong
Senior Scientist, Statistical Programming
Merck & Co., Inc.
yizhuo.zhong@merck.com

Christine Teng
Principal Scientist, Statistical Programming
Merck & Co., Inc.
christine_teng@merck.com

Majdoub Haloui
Principal Scientist, Statistical Programming
Merck & Co., Inc.
majdoub.haloui@merck.com