PharmaSUG China 2025 - Paper CC-190

Development of an Automated Utility for Define.xml Generation and Review in Clinical Data Standardization

Fiona Pan, HUTCHMED Limited

ABSTRACT

Define.xml files are essential metadata documents for clinical trials, ensuring compliance with CDISC standards and regulatory requirements. Current manual processes for creating and reviewing define.xml are labor-intensive, time-consuming, and susceptible to human errors. This paper presents the design and development approach for an automated software tool aimed at streamlining define.xml generation and enhancing review processes in clinical data management workflows.

This paper presents a comprehensive R-based solution for automated generation and validation of define.xml files from ADaM specification sheets. The define.xml file serves as critical metadata documentation in clinical trials, providing essential information about datasets, variables, controlled terminology, and analytical methods. Our methodology demonstrates how to transform Excel-based specifications into compliant define.xml files using custom R functions, followed by rigorous validation processes. The approach includes systematic preparation of specification components, automated generation using the generate_define function, XML conversion through P21 standards, and comprehensive quality assurance using the chk_define function. This streamlined workflow reduces manual effort by approximately 97% while ensuring regulatory compliance and data integrity.

INTRODUCTION

In pharmaceutical research, the define.xml file represents a cornerstone document that provides comprehensive metadata about clinical trial datasets submitted to regulatory authorities. This XML-based specification describes dataset structures, variable definitions, controlled terminologies, and computational methods used throughout the clinical development process.

Traditional manual creation of define.xml files presents significant challenges including time consumption, error susceptibility, and inconsistent formatting. These challenges necessitate automated solutions that can ensure accuracy, efficiency, and regulatory compliance.

This paper addresses these challenges by presenting an integrated R-based framework that transforms structured specification sheets into compliant define.xml files. The methodology emphasizes automation, validation, and quality assurance to meet stringent regulatory requirements while reducing manual workload.

Our approach provides clinical data management teams with tools to:

- Systematically prepare specification metadata
- Automatically generate define.xml files
- Implement comprehensive validation procedures
- Maintain audit trails and documentation standards

DEFINE.XML STRUCTURE AND KEY COMPONENTS

The define.xml file architecture consists of five fundamental components that collectively provide comprehensive metadata documentation for clinical datasets.

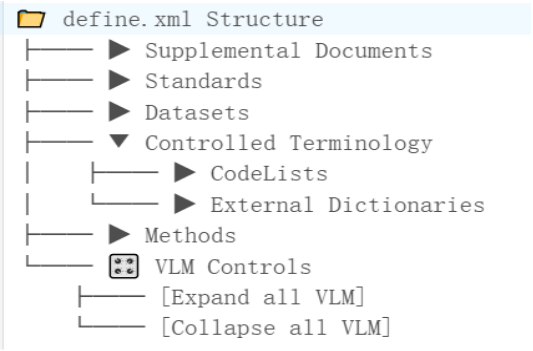

Figure 1. Define.xml Structure

SUPPLEMENTAL DOCUMENTS

The Supplemental Documents section serves as a repository for external documentation that supports data interpretation and regulatory review. This component references analysis data reviewer guides, statistical analysis plans, data collection forms, and other critical documentation that provides context for the clinical datasets.

STANDARDS

The Standards section represents a fundamental component of define.xml that documents all data standards and implementation guides applied throughout the clinical study. This section provides regulatory reviewers and data recipients with essential information about the standardization framework used for data collection, processing, and submission.

DATASETS

The Datasets section forms the structural foundation of define.xml, cataloging all datasets and all datasets variables included in the submission. Each dataset entry includes essential metadata such as dataset names, descriptions, key variables, structural information and Variables, Lable, Type, Length/Format, CT and Origin/Source/Method/Comment.

VARIABLE LEVEL METADATA (VLM)

Variable Level Metadata provides detailed specifications for each variable within the datasets. This includes variable names, labels, data types, formats, lengths, roles, and permissible values. Value-level metadata should be provided when there is a need to describe differing metadata attributes for subsets of cells within a column.

CONTROLLED TERMINOLOGY

The Controlled Terminology section defines standardized vocabularies, code lists, and external dictionaries used throughout the datasets. This component ensures consistent data representation, facilitates data integration across studies, and supports regulatory harmonization efforts. Common terminologies include MedDRA for adverse events, WHO Drug Dictionary for concomitant medications, and custom code lists for study-specific variables.

METHODS

The Methods section documents computational algorithms, derivation rules, and analytical procedures used to create or transform data variables. This critical component provides transparency in data processing, enables reproducibility of analyses, and supports regulatory review of derived endpoints.

COMMENTS

Comments are optional with respect to XML validation.

Component	Primary Purpose	Key Elements
Supplemental Documents	External references	Document titles, locations, descriptions
Standards	Published content standards	SDTM/ADaM CT, SDTM/ADaM IG
Datasets	Structural metadata	Dataset stucture and Each variables information for Datasets
VLM	Subsets of Variable	Variable, XXCAT, PARAMCD
Controlled Terminology	Standardized vocabularies	Code lists, dictionaries, mappings
Methods	Computational algorithms	Derivation rules, formulas

Table 1 summarizes the key characteristics and purposes of each component

ADAM SPECIFICATION PREPARATION

Successful define.xml generation requires systematic preparation of specification sheets with complete and accurate metadata. The preparation process involves configuring multiple interconnected worksheets that collectively define all aspects of the clinical datasets. At HUTCHMED, the ADaM specification structure shares the same framework as the define specification, including Datasets, Variables, ValueLevel, and Codelists sheets, which are already prepared during the ADaM dataset development process. However, compared to the final define specification requirements, additional preparation is needed for the Define, Dictionaries, Methods, Comments, and Documents sheets to complete the comprehensive metadata documentation necessary for regulatory submission.

DEFINE SHEET CONFIGURATION

The Define sheet serves as the master configuration file containing study-level metadata. Essential attributes include:

- StudyName: Unique study identifier
- StudyDescription: Comprehensive study summary
- ProtocolName: Protocol identification
- StandardName: Applied data standard (CDISC, ADaM)
- StandardVersion: Version number of applied standard
- Language: Documentation language code

The content is simple and can be populated manually directly in ADaM spec.

DICTIONARIES SHEET

The Dictionaries sheet defines external terminologies such as MedDRA or WHO Drug Dictionary with version specifications and reference locations. This sheet is also a direct manual entry into the dictionary version.

DOCUMENTS SHEET

The Documents sheet catalogs all supplemental documentation with complete metadata including document titles, descriptions, file locations, and document types. This sheet supports the Supplemental Documents section of define.xml. This sheet is also simple and can be manually filled in beforehand.

VARIABLES AND VALUELEVEL SHEET

Add a DefComment column to contain simplified and updated logic from the ADaM specification. The Methods sheet content presents the algorithm corresponding to variables with Origin=Derived in the Variables and ValueLevel sheets. Some variables may reference external files, therefore EXTERNAL and Epages columns are added to the Variables and ValueLevel sheets to facilitate external file location and referencing.

GENERATE DEFINE SPEC WITH R FUCTION

After preparing everything, use the generate_define function to generate the define spec. The specific logic is: Process each sheet in the define spec, for example, the methods sheet. The methods in the ValueLevel sheet use the following code:

```
vlm_methods <- d_vlm %>%
  filter(ORIGIN=="Derived") %>%
  mutate(
    ID = METHOD,
    NAME = paste0("Algorithm to derive ", trimws(ID)),
    TYPE = "Computation",
    DESCRIPTION = ALGORITHM,
    EXPRESSION_CONTEXT = "",
    EXPRESSION_CODE = "",
    DOCUMENT = EXTERNAL,
    PAGES = EPAGES
  ) %>%
  select(ID, NAME, TYPE, DESCRIPTION, EXPRESSION_CONTEXT,
         EXPRESSION_CODE, DOCUMENT, PAGES)
```

After processing all the required sheets in the define spec, export the content.

```
# Generate comprehensive define specification
data_list <- list(
  "Define" = d_define,
  "Datasets" = d_datasets,
  "Variables" = d_variables,
  "ValueLevel" = d_vlm,
  "Codelists" = d_codelist,
  "Dictionaries" = d_dict,
```

```
  "Methods" = d_methods,
  "Comments" = d_comments,
  "Documents" = d_documents
)
# Export integrated define specification
write.xlsx(data_list, outdefine)
```

Call the generate_define function and generate define spec as below. Then import the generated define spec into the P21 tool to generate define.xml.

```
generate_define(
"path/ADaM spec.xlsx",
"path/define_spec.xlsx"
)
```

VALIDATION LIMITATIONS AND R-BASED SUPPLEMENTAL CHECKS

While automated validation tools provide substantial support for define.xml quality assurance, certain limitations exist that require additional verification approaches, particularly in resource-constrained environments.

P21 COMMUNITY TOOL CAPABILITIES AND LIMITATIONS

P21 Community serves as a widely-used, freely available validation tool that effectively identifies the majority of define.xml compliance issues. This tool successfully validates XML schema adherence, basic cross-reference integrity, and fundamental CDISC standards compliance. However, the free community version has inherent limitations that create validation gaps requiring supplemental verification approaches.

Some critical validation areas are not comprehensively covered by P21 Community due to its free-tier restrictions, such as:

- XPT Length Inconsistencies: P21 Community cannot perform cross-validation between actual dataset (.xpt) file variable lengths and the corresponding length specifications documented in define.xml.
- Origin Source Relationship Validation: The tool lacks comprehensive validation of origin-source relationships, particularly the complex mappings between different origin types and their corresponding source attributions.
- Whether external files have pages, and whether pages exceed the maximum pages of the file.

Type	Source				Notes
	Subject	Investigator	Vendor	Sponsor	
Collected	ePro	CRF	Lab data, ECG	X	This term should be used for clinical data that were actually observed or recorded by a person or received from an instrument; it should not be used for data that have been interpreted, calculated, or derived from other information.
Derived	X	X	Lab data, ECG	SDTM	Derivation examples include calculations performed during data collection (e.g., --DY). Other derivation examples: calculations within ePRO (e.g., questionnaire section scores) and calculations within EDC (e.g., BMI, BSA).
Assigned	X	X	Adjudicator	SDTM	Examples of this include third-party attributions by an adjudicator, coded terms that are supplied as part of a coding process, and values that are set independently of any subject-related data values in order to complete SDTM fields such as DOMAIN and --TESTCD
Protocol	X	X	X	SDTM	An example would be VSPOS (Vital Signs Position), which could be specified in the protocol and be provided by other means (e.g. CRF, eDT).
Predecessor	X	X	X	X	Use when a value is an exact copy of another value in an SDTM dataset.

Table 2 origin-source relationship matrix[1]

R-BASED SUPPLEMENTAL VALIDATION

Given the limited availability of comprehensive paid validation tools in many regional markets, particularly in domestic Chinese submissions, R-based supplemental validation provides a cost-effective solution for addressing P21 Community's limitations.

The following R-based validation functions supplement P21 Community checks :

```
chk_define(
"path/define_spec.xlsx",
"path/CHK_define.xlsx"
)
```

下面是输出结果：

Section	Description
Variables	ADAE STUDYID: Origin is "Predecessor", the Source must be null
Variables	ADAE USUBJID: Origin is "Predecessor", the Source must be null
Variables	ADAE SUBJID: Origin is "Predecessor", the Source must be null
Variables	ADAE SITEID: Origin is "Predecessor", the Source must be null
Variables	ADAE ASTDT: Origin is "Derived", the Source must be "Sponsor"
Variable length	ADAE STUDYID: dataset lengths do not match the lengths stated in the define
Variable length	ADAE USUBJID: dataset lengths do not match the lengths stated in the define
Variable length	ADCV CVAE3: dataset lengths do not match the lengths stated in the define
Variable length	ADEX EXIAE1: dataset lengths do not match the lengths stated in the define

Figure 2. define.xml check output

IMPROVEMENTS IN EFFICIENCY AND QUALITY

R function saved us up to 90% of the time to generate define.xml.

Metric	Manual Process	Automated Process	Improvement
Define.xml Average Creation Time	20 hours	0.5 hours	98% reduction
Validation Errors	24 hours	8 hours	67% reduction

Table 2 Performance Metrics Comparison: Manual vs. Automated Processes

FUTURE ENHANCEMENT OPPORTUNITIES

- Cross check define.xml and review guide
- Check if hyperlinks in define.xml are useful
- ……

CONCLUSION

This comprehensive R-based approach for automated define.xml generation and validation addresses critical challenges in clinical data management by providing efficient, accurate, and compliant solutions for regulatory documentation.

The methodology demonstrates substantial improvements in efficiency, quality, and regulatory compliance compared to traditional manual approaches. Key achievements include 97% reduction in creation time, 67% decrease in validation errors, and comprehensive automation of complex cross-referencing procedures.

The systematic workflow ensures consistent results across studies while maintaining flexibility for study-specific requirements. The comprehensive validation framework provides confidence in regulatory compliance and supports successful submission outcomes.

Implementation of this approach enables clinical data management teams to focus on higher-value analytical activities while ensuring robust, accurate, and compliant define.xml documentation. The scalable architecture supports studies of various sizes and complexities, making it suitable for diverse clinical development programs.

Future enhancements will focus on expanded integration capabilities and advanced analytics to further optimize the define.xml generation process and support evolving regulatory requirements.

REFERENCES

1. CDISC. 2021. "Define-XML Specification Version 2.1." Clinical Data Interchange Standards Consortium. Available at https://www.cdisc.org/standards/data-exchange/define-xml.

CONTACT INFORMATION

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

Fiona Pan
HUTCHMED Limited
fionap@hutch-med.com