

SI06

**Bringing to Light - Analysis Results
Metadata v1.0 for Define-XML v2**

**By,
Limna Salim
Hrishikesh K S
Vani S**

INTRODUCTION

- In January 2015, CDISC published the Analysis Results Metadata , v1.0 (ARM) as an extension to the Define.XML 2.0.0
- Gained less recognition but becoming a necessity for the regulatory submissions.
- Add-in to the ADaM Define.XML
- Metadata that describes the statistical reporting analysis performed by identifying
 - The critical analysis/reason,
 - Links between results,
 - Analysis methods and
 - Programming statements used to generate the statistical result.

Why ARM ?

- Define.XML + ARM facilitates healthy regulatory review
- Traceability.
 - Understand the data flow from collection to the CDISC standard datasets to Analysis results metadata
- Significantly reduces the review time

- + Analysis Data Reviewer's C
- + Statistical Analysis Plan
- + Addendum to SAP v3.1 da
- + Analysis Population Discus
- + Analysis Results Metadata
 - Table T14.2.1.1
 - Table T14.2.1.2
 - Table T14.2.1.3
- + Datasets
 - ADSL (Subject-Level Anal
 - ADCSSRS (CSSRS Analysi
 - ADEFF (MI-Efficacy Analys
 - ADEG (ECG Test Results A
 - ADLB (Laboratory Test Re
 - ADNIV (Nervous System A
 - ADOE (Ophthalmic Examir
 - ADPANS (PANSS Assessm

Analysis Results Metadata - Summary

Table T14.2.1.1 Change from Baseline in Positive and Negative Syndrome Scale (PANSS) Total Score by Visit - MMRM (Modified Intent-to-Treat Population) <u>Inferential statistics including least squares means, standard errors, treatment differences in least squares means, and 95% confidence intervals from MMRM, along with Descriptive Statistics.</u>
Table T14.2.1.2 Sensitivity Analysis on Change from Baseline in Positive and Negative Syndrome Scale (PANSS) Score at Day 29 - ANCOVA Using Multiple Imputation (Modified Intent-to-Treat Population) <u>Inferential statistics including least squares means, standard errors, treatment differences in least squares means, and 95% confidence intervals from ANCOVA using Multiple Imputation , along with Descriptive Statistics.</u>
Table T14.2.1.3 Sensitivity Analysis on Change from Baseline in Positive and Negative Syndrome Scale (PANSS) Total Score at Day 29 - ANCOVA Using WOCF (Modified Intent-to-Treat Population) <u>Inferential statistics including least squares means, standard errors, treatment differences in least squares means, and 95% confidence intervals from ANCOVA using WOCF, along with Descriptive Statistics.</u>

Components of ARM

- The key components in ADaM Analysis Results Metadata are:
 - Analysis Display metadata definitions
 - Analysis Result metadata definitions
 - Analysis parameter(s)
 - Analysis dataset(s)
 - Analysis variable(s)
 - Selection criteria
 - Documentation
 - Programming statements

Analysis Results Metadata - Detail

Table T14.2.1.1

Display	Change from Baseline in Positive and Negative Syndrome Scale (PANSS) Total Score by Visit - MMRM (Modified Intent-to-Treat Population)
Analysis Result	Inferential statistics including least squares means, standard errors, treatment differences in least squares means, and 95% confidence intervals from MMRM, along with Descriptive Statistics.
Analysis Parameter(s)	PARAMCD = "PANSS134 (PANSS1-Total PANSS Score)"
Analysis Variable(s)	ADPANSS_CHG (Change from Baseline)
Analysis Reason	PRIMARY OUTCOME MEASURE
Analysis Purpose	SPECIFIED IN SAP
Data References (incl. Selection Criteria)	ADPANSS [PARAMCD = "PANSS134 (PANSS1-Total PANSS Score)" and MITTFL = "Y" and AVISIT IN ("Day 8", "Day 15", "Day 22", "Day 29") and ANL01FL = "Y"]

Current Industry Approaches

1. Standard Manual updates to current Define.XML
2. Using SAS script
 - Standardized Excel spreadsheet which the study programming team or statisticians can fill out.
 - SAS programs reads in the Excel specification to generate the valid XML code for the define elements specific to the ARM according to Define2.0.
 - Output text file in valid xml syntax manually copied to the ADaM define in proper locations to plug-in the ARM for proper display.
- Drawback
 - Manual implementation into corresponding sections will be difficult for a programmer with no background on XML.
 - More effective hours.

Proposed Automation Framework

- The latest version of P21 still lacks the capability to generate Define.XML with the feature of ARM.
- Python® framework developed which automatically integrates ARM in the Define.XML from P21 for ADaM taking also as input a standardized Excel specification containing all required information for ARM.



Proposed Automation Framework

❖ Step 1-

- The specification follows a similar template and mapping as to the template spec used to generate ADaM Define.XML.
- 4 different tabs filled out correctly for the tool to use the information for analysis results display in the Define.XML

- ARM
- WhereClause
- Comments
- Documents

Table Num	Table page No	Table Description	Analysis result	Dataset Name	Comment ID	Analysis Paramter	Analysis Variable	Analysis Reason	Analysis Purpose	WhereClause ID	Documentation
T14.2.1.1		Change from Baseline in Positive and Negative Syndrome Scale (PANSS) Total Score by Visit - MMRM (Modified Intent-to-Treat Population)	Inferential statistics including least squares means, standard errors, treatment differences in least squares means, and 95% confidence intervals from MMRM, along with Descriptive Statistics.	ADPANSS		ADPANSS.PARAMCD	ADPANSS.CHG	PRIMARY OUTCOME MEASURE	SPECIFIED IN SAP	T14.2.1.1	

ARM

Whereclauses

Comments

Documents

+

Proposed Automation Framework

❖ Step 2 -

- Generation of Define.XML from Pinnacle 21

❖ Step 3 -

- The framework starts by calling in both the inputs available.
- The Python library ElementTree enables parsing and creating XML files.
- Pandas to read in excel spec.

```
from xml.etree.ElementTree import Element, SubElement, Comment, tostring, parse, iterparse, register_namespace
from xml.dom import minidom
import pandas as pd
```

Proposed Automation Framework

❖ Step 3 (Contd.) -

Generate XML code separately for ARM, Where Clause, Comments & leaf/document section and store as Child sections.

```
# Define root element for wc
top = Element('top')
comment = Comment('Generated for easy removal')
top.append(comment)

# Iterate over sheet rows grouped by Table ID
for name, group in dfwc.groupby('ID'):
    whereclause = SubElement(top, 'def:WhereClauseDef')
    whereclause.set('OID', f"WC.{name}")
    for _, row in group.iterrows():
        rangecheck = SubElement(whereclause, 'RangeCheck', {'SoftHard': 'Soft', 'def:ItemOID': f"IT.{row}"})
        for value in row['Value'].split(','):
            checkvalue = SubElement(rangecheck, 'CheckValue')
            checkvalue.text = value

whereclauses = top.find('def:WhereClauseDef')
```

Proposed Automation Framework

❖ Step 3 (Contd.) -

- Parent sections within main Define.XML found by the 'iterparse' function in the ElementTree package.

```
def register_all_namespaces(filename):  
    namespaces = dict([node for _, node in iterparse(filename, events=['start-ns'])])  
    for ns in namespaces:  
        ET.register_namespace(ns, namespaces[ns])
```

- Child sections linked to the Parent sections using the append function.

```
parent.insert(index, whereclauses)  
parent.append(armleaf)  
parent.append(armsection)
```

ARM - Summary

- ARM becoming a necessity for the regulatory submissions
- Traceability
- Need for Standardization
- Straight forward, robust and efficient Python® framework to implement ARM into Define.XML
- Can be implemented with limited tools

Thank You