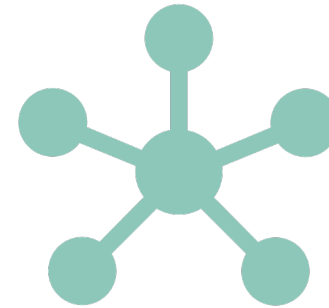


Automating Analysis Results Metadata (ARM)



Ghanshyam Jangid



PHUSE US Connect 2022

Agenda

Analysis Results Metadata overview

Process diagram

Planning and execution

Utility overview

Functional description and execution

Analysis Result metadata as RTF vs Define

Pros and Cons

Matrix

Conclusion

Introduction



Analysis Results Metadata (ARM) is a key component in e-submission analysis package



The ARM document provides traceability for a given analysis result, key information about the analysis method used and the reason the analysis was performed



At Merck, the ARM is generated manually in .docx file and then converted to .pdf



Manual updates can result in human errors, inconsistencies and more time consuming



P21 Enterprise for ARM generation can create some challenges as the ARM is dependent on the completion of the CSR

ARM Document Overview

Study PxxxVxxMKxxxxx

Analysis Results Metadata

Contents

Study PxxxVxxMKxxxxx

Analysis Results Metadata

1. Analysis Define and Data Reviewers Guide

Program & Protocol Number	MK Protocol
Protocol Title	
Analysis Data Definitions	[PxxxVxxMKxxxxx: adam-define]
Analysis Data Reviewers Guide	[PxxxVxxMKxxxxx: adam-adrg]

The Analysis Results Metadata describes the key results and analysis dataset found in the analysis\adam subfolder. The ADaM dataset is the source data used to support the analysis.

ARM Document Overview – Section 3

Study PxxxVxxMKxxxx

Analysis Results Metadata

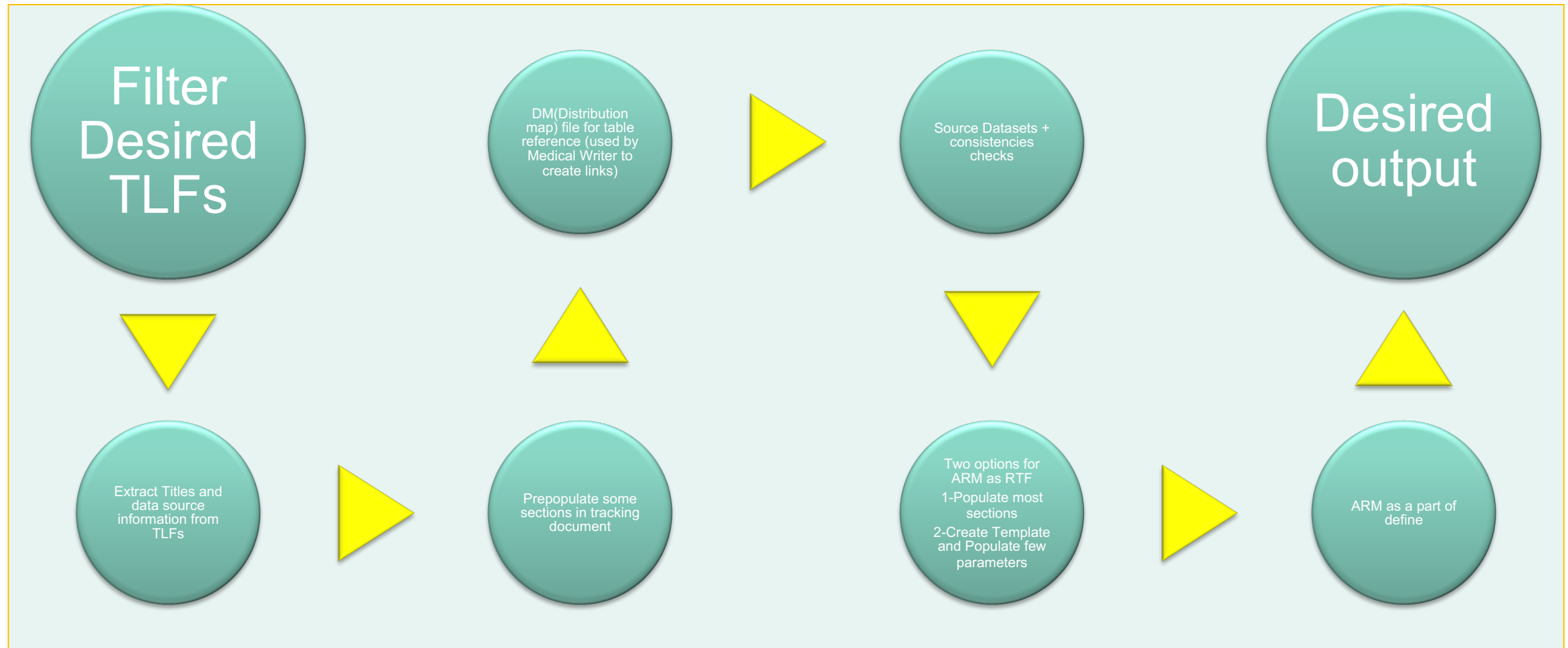
2. Analysis Results Metadata Summary

Table Reference	Table Title	SAS Program Name (programs)	Input File Name	
			Analysis (datasets)	Tabulations (legacy)
[Refer to Citation Construct Job Aid] <i>Citations in square brackets cannot straddle two pages</i>	Disposition of Subjects	call0pr0disposition.txt (callid=2) <i>Do not wrap filename</i>	adsl.xpt <i>Use ENTER after each dataset. Do not wrap filename</i>	dsplus.xpt explus.xpt <i>Use ENTER after each dataset. Do not wrap filename</i>

3. Analysis Results Metadata Details

Table Reference: [Refer to Citation Construct Job Aid] <i>NOTE: Citations in square brackets cannot straddle two pages and cannot contain carriage return.</i>	
Analysis Result	<i><include analysis result of interest></i>
Analysis Parameter (s)	<i><add useful parameters that is the focus of the analysis result></i>
Analysis Variable (s)	<i><include specific variables that will be analyzed></i>
Analysis Reason	<i><define the reason for including the analysis></i>
Analysis Purpose	<i><define the purpose of the analysis></i>
Data References (incl. Selection Criteria)	<i><add selection criteria that describes the analysis subset used to derive a result for a display></i>
Documentation	<i><Optional section intended to guide reviewers to external references such as protocol></i>
Programming Statements	<i><add program code used to perform the specific analysis and show how the specific analysis result was created></i>

Process Diagram



Planning and Execution

- Multiple source files are needed to create Analysis Results Metadata
- For example: ADaM metadata, SAS programs, TLF specification, SAP etc.
- Identify the list of outputs needed to create Analysis Results Metadata
- Compare the list with the folder contains TLFs
- Extract titles and data source information from TLFs using a utility macro
- Follow standard text/template to ensure consistency
- Statistician input as needed for relevant sections

Planning and Execution(cont.)

- Collect all information from different source (RTF, DM file, tracking file)
- Perform consistency checks between collected information vs actual ADAM datasets
- Add two options
 1. Analysis Results Metadata as part of RTF(RTF to PDF)
 2. Analysis Results Metadata as part of Define

Utility call information

```
/**Example-with Project tracking*/
```

```
%arm0esub
```

```
(ansel    = Y  
,anrlocation = %str(&protpath)  
,anrname    = %str(Project_tracking.xlsx)  
,anrsheet   = %str(TLFs)  
,tabfilref  = %str(fptotb)  
,figfilref  = %str(phocg)  
,ptitle     = %str(Protocol Title)  
,dmfile     = DM_table_ref.xlsx  
,dmsheet    = DM  
,outputlink = %str(ARM_location)  
,output     = ARM_file  
)
```

Functional Description

- Utility reads project tracking and select required output file names for ARM
- Extract titles and dataset source information from TLFs
- To generate certain sections in ARM, project tracking sheet needs to be updated (for example: parameter used, Analysis variable, Statistical models etc.)
- Compare with ADAM datasets to ensure consistencies between ARM vs ADaM datasets
- Used proc report to create same layout and format as standard template
- Added two options to generate ARM files
 1. ARM as RTF (convert to PDFs)
 2. ARM as a part of define

Execution

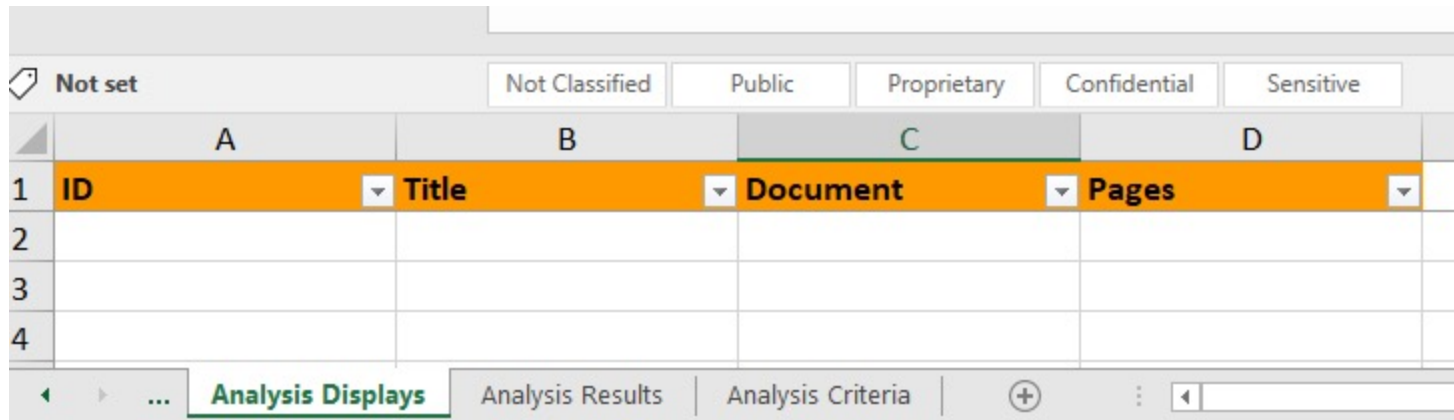
- Project tracking must be accurate with the following information
 - Identifying flag for ARM
 - TLFs output file names
- TLFs location
- Add protocol title in calling macro
- DM file
- Final ARM location

ARM file as RTF using Proc Report

- Information collected from different files is split into SAS datasets for the 3 ARM sections
- Add multiple proc report calls to match standard template layout
- Added blank page for TOC
- TOC will be created manually

ARM as a part of Define

- Export define as excel
- Auto populate below three tables(specific for ARM: Analysis Displays, Analysis Results, Analysis Criteria) after collecting all required information using a utility
- Reimport define to P21 Enterprise in order to see ARM as part of define (in Screenshot 2)



The screenshot shows a software interface with a table. The table has four columns: ID, Title, Document, and Pages. The first row is highlighted in orange. Below the table, there are tabs for 'Analysis Displays', 'Analysis Results', and 'Analysis Criteria'. The 'Analysis Displays' tab is currently selected. The interface also includes a header section with a 'Not set' label and several buttons: 'Not Classified', 'Public', 'Proprietary', 'Confidential', and 'Sensitive'.

	A	B	C	D
1	ID	Title	Document	Pages
2				
3				
4				

Screenshot: 1

Example of ARM as a part of define

ARM after reimport of updated excel file into P21 Enterprise

Analysis Reviewer's Guide
Analysis-Result-Metadata

- ▶ Analysis Datasets
- ▶ Parameter Value Level Metadata
- ▶ Controlled Terminology
- ▶ Analysis Derivations
- ▶ Comments

Stylesheet version: 2015-01-16

Standard	ADaM-IG 1.0
Study Name	P000V00MK0000
Study Description	MK0000-000: A Clinical Trial of Pembrolizumab (MK-0000) Evaluating Predictive Biomarkers in Subjects with Advanced Solid Tumors (KEYNOTE 000)
Protocol Name	P000V00MK0000

Analysis Datasets for Study P000V00MK0000 (ADaM-IG 1.0)

Dataset	Description	Class	Structure	Purpose	Keys	Location	Documentation
ADSL	Subject-Level Analysis Dataset	SUBJECT LEVEL ANALYSIS DATASET	One record per subject	Analysis	USUBJID	adsl.xpt	
ADDILI	DILI Reporting Analysis Dataset	BASIC DATA STRUCTURE	One observation per subject, parameter, and analysis time point.	Analysis	USUBJID, PARAMCD, AEPOCH, ADT	addili.xpt	

Screenshot: 2

Matrix

Manual process	Auto populate sections with utility
Manual update of entire document	Automate generation of most sections
Manually adding titles, source Dataset information, Analysis parameters, Analysis variables etc. can be significantly time consuming	Dynamically find all required information and auto populate
Possible human error such as typo, space, layout etc. when generating ARM	Human error can be significantly minimized
Estimated time 1-2 days	Estimated time 3-5 hours
May lead to inconsistencies across different submissions(example: Format, layout ,typos in files etc.)	Maintain consistencies across all studies

Matrix: Pros and Cons (ARM as rtf)

Pros

- Automate generating most sections in ARM document (rtf->pdf)
- Majority of the information can be obtained dynamically for ARM and populate automatically
- Reduce the level of manual QC and human errors
- Estimated time to create ARM is 3-5 hours compare to manual update
- Achieve consistencies with SOP/guidance and across studies

Cons/limitations

- Project tracker needs to be up to date before an accurate ARM can be generated

Pros and Cons (ARM as Define)

Pros

- Automate generating most sections in ARM document (xml)
- Majority of the information can be obtained dynamically for ARM and populate automatically
- Reduce the level of manual QC and human errors
- Estimated time to create ARM in 3-5 hours
- Maintain consistencies with SOP/guidance and across studies

Cons/limitations

- Project tracker needs to be up to date before an accurate ARM can be generated
- Generating ARM as part of define.xml:
- Path to the actual CSR in the eCTD folder structure is not available until regulatory starts their publishing work
- ARM is not finalized until the CSR is Final and locked

Conclusion

We can significantly reduce development cycle of Analysis results metadata with significant improvement in quality and achieve higher efficiencies

