

## Doctor's 'Prescription' to Re-engineer Process of Pinnacle 21 Community Version Friendly ADaM Development

Aakar Shah, Pfizer Inc; Tracy Sherman, Efficacy Consulting Group, Inc.

### ABSTRACT

With development and availability of Pinnacle 21 community version software, a lot of power has been given back into the hands of SAS® programmer at any skill level to generate Define.xml 2.0 and to validate ADaM used for submission. The new prescriptive approach of generating Define.xml upfront creates an opportunity to not just modify but re-engineer the ADaM development process by either using Define specifications as programming specifications, ADaM attribute assignments and/or data validation. It also improves communication within the programming team and with stakeholders such as the study statistician for reviewing ADaM specification throughout the study life cycle.

This paper will outline the re-engineered process of ADaM development including step-by-step instructions to complete Define.xml specifications used by Pinnacle 21 Community software. We will also include details of how to use Pinnacle 21 Define specifications as programming specifications, attribute macro requirements, and will identify supporting utility development.

### INTRODUCTION

The traditional way of ADaM development includes creation of ADaM specifications during study set up using sponsor standard specification template. ADaM datasets are created and maintained throughout the study. After study completion, Define.xml is created using completed set of ADaM datasets as a starting point. This process of creating Define.xml is also called descriptive approach as the Define.xml describes the data you already have. In this popular approach, the datasets are scanned for metadata extraction from the ADaM datasets into a spreadsheet. The user then fills out derivations and other missing information using ADaM specifications as well as ADaM datasets

This traditional way of ADaM creation has many short comings. The whole process is not at all efficient as you are essentially writing the specifications twice, first at the beginning of the study as ADaM specification and later filling out missing information in the Define specification file after final ADaM are created and metadata has been extracted. You also need to keep both specification files aligned in case there are changes to the derivations.

The re-engineered ADaM development process includes creating Define.xml in the beginning by defining datasets, variables, derivations, other metadata, and codelists during study set up. You can then use the Define specification as the programming specifications, ADaM attribute assignments and to validate incoming study data. This approach is also called prescriptive approach as it prescribes or recommends actions and outcomes throughout the life cycle of the study.

### THE RE-ENGINEERED PROCESS

The overview of re-engineered ADaM development process is shown in Figure 1. Define specifications should be created at the beginning of the ADaM development, not at the end. The user starts with the Define specification template to create ADaM specifications as explained in detail in the later section. You could also modify the template in certain way as explained in the later section to make it more user-friendly. The Pinnacle 21 Define generator will directly use the specification file to create the Define.xml. Note that you do not need the ADaM datasets to create the Define.xml. The Statistician or other reviewer could just use the Define.xml to review the ADaM specifications.

The attribute macro uses the define specification spreadsheet to assign attributes in certain ways and to add selected core variables to ADaM datasets from ADSL. The paper includes detailed macro requirements which you could use to create the attribute macro needed for this process. The paper also identifies several utilities you could build to automate Define specification writing.

Once Define.xml and ADaM datasets are available, you can run the Pinnacle 21 validator to generate the Pinnacle 21 validation report. The Pinnacle 21 validation report could be used in different ways. You can update Define specifications and ADaM programs per the validator feedback. You can also use the report to check quality of incoming data by reviewing findings and providing feedback to the SDTM programmer and data manager. Having the Define.xml file available in the beginning, it is recommended to run the validator every time you receive a new set of SDTM. The whole process is quicker since all pieces (ADaM datasets and Define.xml) are available.

At the time of submission, you will save time as there is no need to re-create the specifications for Define.xml as it is available from the beginning.

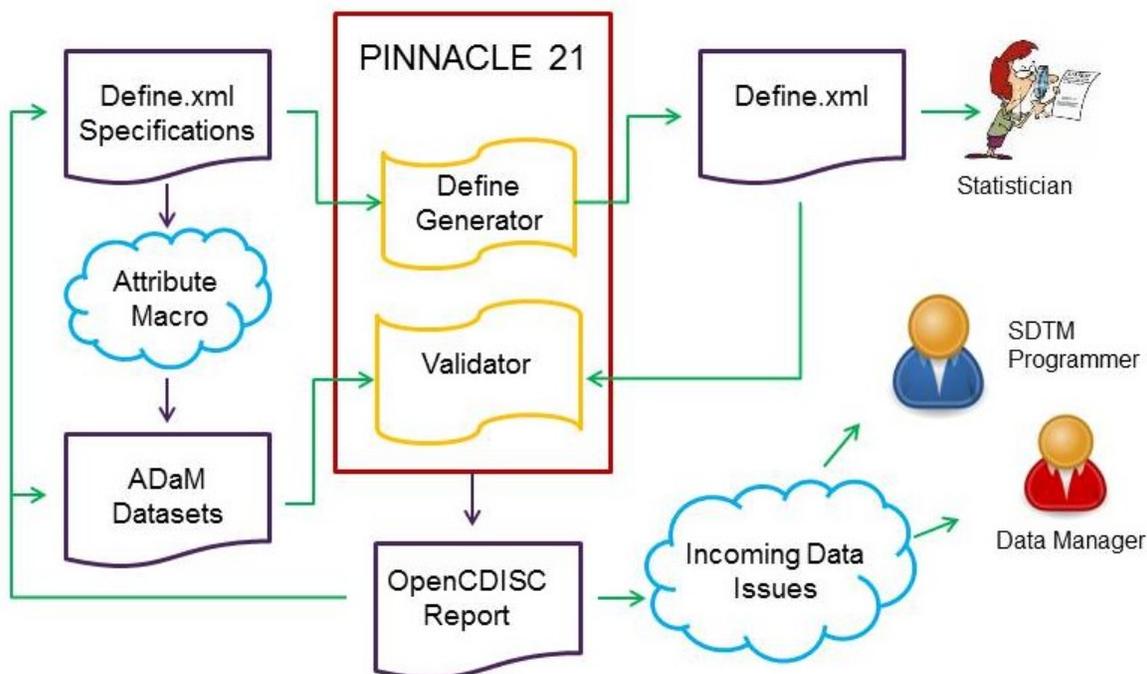


Figure 1 Re-engineered ADaM Development Process

## COMPLETE GUIDE ON HOW TO WRITE DEFINE.XML SPECIFICATION AND MODIFY THE TEMPLATE TO USE FOR PROGRAMMING SPECIFICATION

You can download the Pinnacle 21 Community software for no cost at <https://www.pinnacle21.com/downloads>. If you do not have a copy of the Define specification spreadsheet used by Pinnacle 21, you could easily get it from the community version software by uploading any ADaM dataset. For more details on creating the specifications from source data please refer to <https://www.pinnacle21.com/projects/using-opencdisc-community>.

The Define specification workbook contains 10 worksheets: Study, Datasets, Variables, Valuelevel, Whereclauses, Codelists, Dictionaries, Methods, Comments, and Documents. Below is the detailed description of each worksheet of the Define specification spreadsheet and guidance for populating each column.

**WORKSHEET NAME: STUDY**

<b>Worksheet Column</b>	<b>Guidance</b>
Attribute	Define specification needs following 6 attributes. StudyName StudyDescription ProtocolName StandardName StandardVersion Language
Value	Value for the 6 attributes. StudyName - Enter Name of the Study. This text will become the part of the Define.xml file name. This will show up at the top of define.xml file. StudyDescription - This will show up at the top of define.xml file. ProtocolName - This will show up in the beginning of the define.xml file. StandardName - Default value "ADaM-IG". This will show up at the top of define.xml file. StandardVersion - Default value "1.0". This will show up at the top of the define.xml file. Language - Default Value "en". This will show up at the top of define.xml file.

**WORKSHEET NAME: DATASETS**

<b>Worksheet Column</b>	<b>Guidance</b>
Dataset	Dataset must be the same as SAS Dataset Name but without the ".xpt" extension.
Description	Description of Dataset should exactly match the label for the relevant standard dataset. For custom domains or datasets, provide a short description of the type of data contained within the dataset.
Class	For regulatory submissions of ADaM datasets, a standard order of display has not been established. However, the following Class order seems reasonable, as it is consistent with the ordering of Class values in ADaM 2.1: * SUBJECT LEVEL ANALYSIS DATASET * OCCURRENCE DATA STRUCTURE (For ADVERSE EVENTS ANALYSIS DATASET only) * BASIC DATA STRUCTURE and Other OCCURRENCE DATA STRUCTURE-datasets in alphabetical order by dataset name * ADAM OTHER Note: Use UPPERCASE.
Structure	Description of the level of detail represented by individual records in the dataset. E.g. ADQSADAS dataset: "One record per subject per parameter per analysis visit per analysis date"
Purpose	Use "Analysis"
Key Variables	e.g. STUDYID,USUBJID,AEDECOD,AESEQ; STUDYID,USUBJID,PARAM,PARAMCD
Repeating	"Yes" or "No". Indicates whether a domain contains more than one record per subject or only one record per subject. Set Repeating="No" when used in the reference data section
Reference Data	"Yes" or "No". Indicates whether the dataset contains reference data (not subject data)

Worksheet Column	Guidance
Comment	If the dataset has related comment (text, link to external file etc.), assign the comment ID as <dataset name in uppercase>; Datasets.Comment=Comments.ID

## WORKSHEET NAME: VARIABLES

Worksheet Column	Guidance
Order	Order in which the variable appears in the dataset
Dataset	Dataset must be the same as SAS Dataset Name but without the ".xpt" extension. Use UPPERCASE
Variable	Name of variables. Use UPPERCASE
Label	Variable Label
Data Type	"text", "float", "integer", "date", "datetime", 'time'. The data type of the variable or value. "date", 'time" and "datetime" is used for character date for date/time values in ISO 8601 format. "Integer" is used for numeric date. Assign "text" for partial date.
Length	Variable length. Do not fill. Length should be defined as the maximum expected variable length. Should only be present for DataType equal to "text", "integer", or "float". Develop a utility program to populate this column.
Significant Digit	Do not fill. Required if DataType is "float". Develop a utility program to populate this column.
Format	For Numeric Date ("Integer" Type) - use "date9.", "time5.". Do not fill for rest.
Mandatory	From ADaM IG. "Yes" for required variables.
Codelist	Assign "ISO8601" for character date variables with type "date", "datetime", "time". Assign external dictionary IDs (e.g. AEDICT, DRUGDICT) and link it to corresponding item in "Dictionaries" sheet. Else Variables.Codelist=Codelists.ID or Variables.Codelist=Dictionaries.ID. Use UPPERCASE.
Origin	<p>* Predecessor - Data that is copied from a variable in another dataset. For example, predecessor is used to link ADaM data back to SDTM variables or other ADaM variables to establish traceability. If selected, Define will include text from "Predecessor" column.</p> <p>* Derived- Data that is not directly collected on the CRF or received via eDT, but is calculated by an algorithm or reproducible rule defined by the sponsor, which is dependent upon other data values. Use with "Method" column.</p> <p>* Assigned - Data that is determined by individual judgment (by an evaluator other than the subject or investigator), rather than collected as part of the CRF, eDT or derived based on an algorithm. This may include third party attributions by an adjudicator. Coded terms that are supplied as part of a coding process (as in -- DECOD) are considered to have an Origin of "Assigned". Most of the coded terms, such as PARAMCD, AVISITN, DTYPE, are "Assigned" in source. Use with "Comment" column and corresponding ID in "Comments" sheet to add more details if needed.</p> <p>* eDT - Data that is received via an electronic Data Transfer (eDT). Use with "Comment" column and corresponding ID in "Comments" sheet.</p> <p>*&lt;Missing&gt; if more than one origin for value level data</p>
Pages	Do not fill

<b>Worksheet Column</b>	<b>Guidance</b>
Method	MT.<ADaM Name>.<Variable Name>. If a method can be used across multiple domain then MT.<Name>. Variables.Method = Methods.ID. Use when origin is "Derived". Use UPPERCASE
Predecessor	<SDTM>.<Variable Name> or <ADaM>.<Variable Name>. Use UPPERCASE
Role	Do not fill.
Comment	<ADaM>.<Variable Name>. Variables.Comment=Comments.ID. Use UPPERCASE

### WORKSHEET NAME: VALUELEVEL

<b>Worksheet Column</b>	<b>Guidance</b>
Order	Do not fill.
Dataset	Dataset Name
Variable	Variable Name
Where Clause	ValueLevel.Where Clause = WhereClauses.ID. You may create several records in WhereClause sheet for a "multiple" condition.
Data Type	Value Level Data Type.
Length	Refer to "Variables" sheet guidance
Significant Digit	Refer to "Variables" sheet guidance
Format	Refer to "Variables" sheet guidance
Mandatory	Do not fill
Codelist	A utility can be developed to fill up the column. Variables.Codelist=ValueLevel.Codelist for the same variable of a dataset
Origin	Refer to "Variables" sheet guidance
Pages	Do not fill
Method	Refer to "Variables" sheet guidance
Predecessor	Refer to "Variables" sheet guidance
Value Level Comment	<ADaM>.<Variable Name>.<Value> as ID to link "Comments" sheet. Use UPPERCASE
Join Comment	Do not fill.

### WORKSHEET NAME: WHERECLAUSES

<b>Worksheet Column</b>	<b>Guidance</b>
ID	ValueLevel.Where Clause = WhereClauses.ID. It may be several rows for a "multiple" condition
Dataset	Dataset Name
Variable	Variable Name
Comparator	EQ, NE, IN, NOTIN, LT, LE, GT, GE
Value	One or multiple values from the "Variable"

## WORKSHEET NAME: CODELISTS

Worksheet Column	Guidance
ID	Variables.Codelist=Codelists.ID or ValueLevel.Codelist=Codelists.ID. Use UPPERCASE. ID will be printed in "Controlled Terminology" section of the Define.xml. Add rows with same ID for each Term of the variable.
Name	Name of Codelist
NCI Codelist Code	Enter published NCI Codelist Code.
Data Type	Enter Data Type of the variable linked to the ID.
Order	Ordering of the values of controlled terminology as printed on Define.xml.
Term	Values from ADaM variable. One row per term.
NCI Term Code	Enter published NCI Term Code. Develop a utility program to populate this column using CDISC published terminology. If certain codes are missing within a codelist then system will assume that those code is part of sponsor defined extended list.
Decoded Value	Enter Decoded value.

## WORKSHEET NAME: DICTIONARIES

Worksheet Column	Guidance
ID	Link ID to the "Codelist" column in "Variables" or "ValueLevel" sheet. E.g. AEDICT, DRUGDICT. Use UPPERCASE
Name	Text for the dictionary which will be printed on Define under Codelist column. E.g. "Adverse Event Dictionary", "Drug Dictionary".
Data Type	Data type of the linked variable
Dictionary	Name of the dictionary. E.g. "MedDRA", "WHODRUG"
Version	Version of the dictionary

## WORKSHEET NAME: METHODS

Worksheet Column	Guidance
ID	MT.<ADaM Name>.<Variable Name> as ID where Variables.Method=Methods.ID; MT. OR <ADaM Name>.<Variable Name>.<Value> as ID where ValueLevel.Method=Methods.ID. Use UPPERCASE
Name	Same as Methods.ID
Type	Assign "Computation" or "Imputation".
Description	Description of the derivation.
Expression Context	Do not fill
Expression Code	Do not fill
Document	ID for the external document link to "Documents" sheet.Methods.Document=Documents.ID
Pages	Do not fill

## WORKSHEET NAME: COMMENTS

Worksheet Column	Guidance
ID	Link ID to the "Comment" columns in various sheet
Description	Description of the comment.
Document	ID for the external document link to "Documents" sheet.
Pages	Do not fill

## WORKSHEET NAME: DOCUMENTS

Worksheet Column	Guidance
ID	Link ID to Comments.Document
Title	Text for the document to be printed on the define
Href	Hyperlink to the external document. To link file in the same folder as define.xml, provide name of the file with extension, e.g. analysis-data-reviewers-guide.pdf. To link specific section of the external PDF document, create a destination within PDF and then use "#" to link the destination, e.g. analysis-data-reviewers-guide.pdf#Section5.2.1. To link file located in different folder than the define.xml, provide relative path, e.g. ".\programs\adae-sas.txt".

## HOW TO MODIFY THE STANDARD DEFINE.XML SPECIFICATION FILE TO USE AS PROGRAMMING SPECIFICATIONS

If the order of the worksheets in the Define specification spreadsheet has changed or the order of the columns in any of the worksheets has changed, Pinnacle 21 Community Version Software will not be able to read the workbook. However you can add new worksheets at the end of the pre-defined ten worksheets or add any number of columns after the pre-defined columns in each worksheet.

On the 'Variables' worksheet, instead of repeating the core variables from ADSL that are used in all other datasets, add a column "Core Variable" with 'Y' next to those variables for ADSL records only. Usual Excel tricks could be implemented to create filters, drop down lists for data entry, conditional highlighting for missing cells, etc.

Adding an additional worksheet such as 'ADRG Notes' for documenting issues that may require for explaining in the ADRG (analysis data reviewer's guide), is always helpful. You may also add a worksheet such as 'Discussion' where ongoing data and programming issues were tracked and resolved.

Be sure to check that the worksheets are not filtered when you upload the specifications to Community as you will lose those records that are not visible.

## ATTRIBUTE MACRO REQUIREMENTS

The purpose of the attribute macro is to assign attributes such as order, label, length, significant digits, and formats, per the ADaM specifications. The following requirements specifically work with the Define specification worksheet used by Pinnacle21 Community software:

1. Assign dataset label per the specification – Datasets worksheet.
2. Maintain variable order per the specification - Variables worksheet.
3. Assign variable name per the specification (in uppercase) - Variables worksheet.
4. Assign variable label per the specification – Variables worksheet.

5. Assign variable length per the specification - Variables worksheet.
  - a. If Data Type = integer, float – assign length =8
  - b. If Data Type = text – assign length = \$200
  - c. If Data Type = date, datetime, time – assign length = \$19
6. Remove existing formats from input data
7. Remove existing informats from input data
8. Assign variable format per the specification - Variables worksheet.
9. Output parameter to allow libname.dataset value
10. If macro call indicates to attach “Core variables”
  - a. The macro reads in ADSL specification from Variables worksheet to grab core variables indicated as “Y” in column “Core Variable”;
  - b. Merge selected variables in step 10a from ADSL with the input data by USUBJID.
  - c. Maintains variable order of ADSL before adding variables from the input dataset
11. If a variable mentioned in the specification but not present in the input dataset
  - a. Macro to output variable with missing values in the output dataset
  - b. Provide a list in the log of such variables
12. Sort the output dataset per “Key Variables” column in Datasets worksheet  
Macro to provide info in the log if the sorting key variables cannot identify unique rows.

## CONCLUSION

As demonstrated, the re-engineered process of ADaM development for creating Define.xml at the beginning of study set up results in an efficient, streamlined process resulting in high quality deliverables. This process makes it easier for statisticians to review ADaM specifications and datasets by viewing the Define.xml in their browser. Programmers will find that by using this process will help validate incoming data during the life cycle of a study. Earlier feedback from the Pinnacle 21 validation report can be provided to the SDTM team or data managers.

We recommend that ADaM specification writing is started using the define specification template with the proposed step-by-step guidelines. An attribute macro could be created using the requirements provided in this paper to streamline the process of completing the define specifications.

Many pharmaceutical companies outsource the Define.xml development to outside clinical research organizations. We recommend that the specification template and the validation tools provided within Pinnacle 21 Community be used to provide these deliverables at the earliest possible point. The completed specifications would be shared with the team so that Define.xml modifications and independent validation could be ongoing during the study development life cycle. The earlier the re-engineered process begins, the more time is saved and the higher the quality of data.

## REFERENCES

1. Sirichenko, Sergiy, DiGiantomasso, Michael and Collopy, Travis. 2015. “Usage of OpenCDISC Community Toolset 2.0 for Clinical Programmers”. PharmaSUG Conference. HT04.
2. <https://www.pinnacle21.com/>

## ACKNOWLEDGMENTS

We would like to thank Shuxuan Zhao from Pfizer Inc. and Ganesh Gopal from Epicacy Consulting Group for their continued support and encouragement, as well as all our family, friends and colleagues.

## RECOMMENDED READING

1. Yan, Lin, "Developing ADaM Specifications to Embrace Define-XML 2.0.0 Requirements", 2014, PharmaSUG Conference, DS10

## CONTACT INFORMATION

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

Name: Aakar Shah  
Enterprise: Pfizer, Inc.  
E-mail: [Aakar.Shah@pfizer.com](mailto:Aakar.Shah@pfizer.com)  
[www.pfizer.com](http://www.pfizer.com)

Name: Tracy Sherman  
Enterprise: Efficacy Consulting Group, Inc.  
E-mail: [shermantracy@gmail.com](mailto:shermantracy@gmail.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.