Data Integrity through DEFINE.PDF and DEFINE.XML

Sy Truong, Meta-Xceed, Inc, Fremont, CA

ABSTRACT

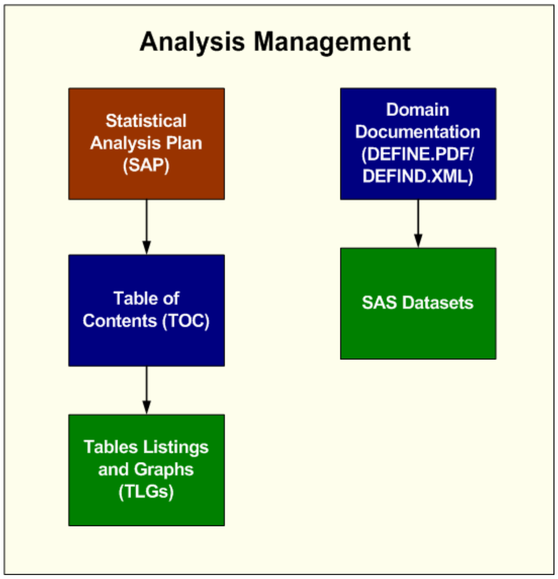
One of the key questions asked in determining if an analysis dataset is valid is simply, what did you do to get to those derived variables? This simple question is what a reviewer would ask during an audit as in a regulatory audit of an electronic submission. The answers to this question are found in the documentation of the analysis dataset. This can be found in the define.pdf or define.xml files. This paper will explore the steps needed in documenting this metadata along with methodologies and utilities to automate it. Some of the topics covered include:

1. Organizing all the datasets that are to be documented
2. Documenting variable attributes as seen in a PROC CONTENTS
3. Decoding format codes
4. Determining the origins or source of the data
5. Ways of automating the capture and presentation of the metadata

In an environment where regulatory requirements change, the need to document how variables are derived still remain the essence to understanding the data. It may be a tedious step to perform the documentation, but with the proper approach and tools, this can be performed with great accuracy and becomes less of a headache.

INTRODUCTION

The main task in the job as a "knowledge worker" is to manage and analyze information. This information can be in the domain of financial data, health care or clinical trials. The increased volume and complexity of the data requires better methodologies and technologies. In the past, the focus of using SAS to analyze clinical trials data has been to generate reports to confirm an analysis. Examples include analysis for adverse event summaries or subject demographic listings. As the number of summary tables, listings and graphs (TLGs) gets larger, it becomes necessary that a statistical analysis plan (SAP) with a table of contents (TOC) be used to manage the analysis. The table of contents lists all the reports needed in a particular study including mockups of how each report is to be used. The biostatistician and SAS programmer work together in creating the reports with the SAP and TOC as a road map to coordinate the effort.


Analysis Management
Statistical Analysis Plan (SAP)
Table of Contents (TOC)
Tables Listings and Graphs (TLGs)
Domain Documentation (DEFINE.PDF/ DEFIND.XML)
SAS Datasets

The reports are still critical to the submission. However, there has been a shift in focus to the accompanying data. Rather than reviewing just the TLGs, FDA reviewers are looking at the data sets. As analogous to the TLGs, the increase in volume and complexity of the data makes it necessary for more effective tools and methodologies. One such methodology is the use of a domain documentation or better known as DEFINE.PDF. This documents what datasets are included with additional metadata such as the variable attributes and whether it is derived or a source variable from the Case Report Form (CRF). This method of organization helps a reviewer navigate and understand the data but it can also act as a road map for SAS programmers and biostatisticians in a similar way that the SAP and TOC help to organize the development of TLGs.

The data domain documentation started as a PDF document named DEFINE.PDF which is part of an electronic submission. The standards and technologies evolved to make the process more universal so rather than using PDF, XML is being adopted. Future electronic submissions will therefore include a DEFINE.XML rather than DEFINE.PDF. In either case, the value of organizing the metadata remains the same. The domain documentation remains an effective tool which provides a guide for those working on creating the data yet it also acts as a roadmap to help the reviewers more effectively navigate through the data.

USING DOMAIN DOCUMENTATION

FDA reviewers use the DEFINE.PDF/DEFINE.XML as a road map to navigate and understand the data that accompanies an electronic submission. In addition to this objective, the domain documentation can have other uses for SAS programmers and biostatisticians who work on creating these data. Some of the uses include:

- Project Management – Organize the target deliverables for SAS data and their status as it is being developed.
- Data Standards – Maintain data structure standards internally or with industry standards such as CDISC.
- Data Integrity and Review – Data quality review and audit in a similar way that a FDA reviewer would perform.

You are not limited to these specific uses but these are examples of how useful this documentation can be within your development team.

PROJECT MANAGEMENT

The most efficient way of managing the data to be included in an electronic submission is to have a clear vision of what that is going to look like. Rather than waiting until the very end to perform documentation, you can use this documentation as a guiding roadmap to what your efforts will be. This will eliminate duplicate and unnecessary wasted efforts since you will only focus on developing data that is going to be included in the final target submission. The following steps describe how you can create this documentation as a project management tool. This can be either in MSWord or Excel format but the following example below shows it in MSWord.

STEP 1: Generate a list of all the datasets that are to be included in your target electronic submission.

Dataset Name	Location	Keys	Number of Variables	Number of Records	Comment
ae	ae.xpt	usubjid	9	20	
cm	cm.xpt	usubjid	4	35	
co	co.xpt	usubjid	15	42	
dm	dm.xpt	usubjid	7	20	
ds	ds.xpt	usubjid	5	20	
ex	ex.xpt	usubjid	7	20	
relrec	relrec.xpt	usubjid	9	251	
su	su.xpt	usubjid	7	80	
suppqual	suppqual.xpt	usubjid	13	2102	

The main objective here is to know what exactly is going to be included. The first significant column for the purpose of project management is the "Dataset Name" column. The linking to transport files is not significant. The keys are usually synonymous to how the dataset is to be sorted so this is also helpful information. The other details are not as important.

The purpose of the comment is a free text field that can be used to describe the dataset. This is a dynamic field that can constantly change to reflect updates to the data as it is being developed. For example, it could start out with the value of "Initial Draft, June 16, 2005". This would then be updated to reflect the status. Some examples of the comment could be:

1. Initial Draft, June 16, 2005 by ST
2. New derivation for age and weight unit calculations were added, June 17, 2005 by SW
3. Verification testing confirming calculations completed, June 20, 2005, by JD
4. Demographic data set with additional derivations for age and weight, July 21, 2005, by ST

The comments can be useful to track at the dataset level how the data is being developed. As it nears the end, the final comment can be reflective of the data for the reviewer as intended for the final review purposes.

STEP 2: Document what variables and associated attributes are going to be included.

ae (ae.xpt)								
Variable Name	**Type**	**Length**	**Variable Label**	**Format**	**Decode Formats**	**Origins**	**Role**	**Comment**
aeacn	Character	100	Action Taken with Study Treatment	actfmt.	1 = None 2 = Dose Increase 3 = Dose Decrease	Derived		
aeendy	Numeric	8	Study Day of End of Event			Derived		
Aesrc	Character	100	Adverse Event Collection Source			Derived		
aestdtc	Character	100	Start Date/Time of Adverse Event			Derived		
aestdy	Numeric	8	Study Day of Start of Event			Derived		
aeterm	Character	100	Reported Term for the Adverse Event			Derived		
siteid	Character	100	Study Site Identifier			Derived		
studyid	Character	100	Study Identifier			Derived		
usubjid	Character	100	Unique Subject Identifier			Derived	Key	

Before any programming starts, it is useful to document all the variables that are to be included in each dataset. This can then drive the programming and determine what variables are going to be derived. This portion of the documentation can change during the design phase. Variable names can change and formats can also change, among other attributes. This is intended to be the target and can be used as part of the dataset specifications in the statistical analysis plan (SAP). Each column can contribute to the design and implementation during project management.

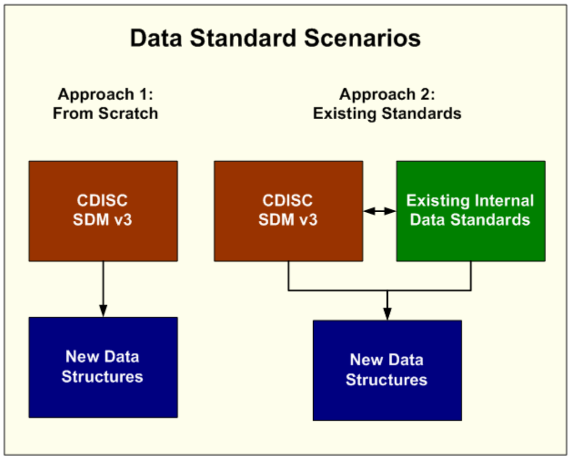
1. Variable Name – The decision to name a variable can come from the existing source dataset or standards that exist internal to your organization or industry standards such as CDISC. This documents what variables will be used during design and throughout the project development.
2. Type / Length – The decision of variable type and length are decided during the planning stages of the project prior to any coding. This will help with standards and consistent targets.
3. Variable Label – These labels can also drive headers in some listings and summary tables so it is important to develop these at the beginning of the project. Modifications during the project can be updated and managed by this document.
4. Format / Decode Formats – This is a useful design tool documenting how certain reports will display coded values. This can be established at the beginning of the project for effective programming.

5. Origins – Variables can come from either the source case report form or derived. If they are derived, you can create a hyperlink to a description of the derivation to help manage the code that is going to be developed.
6. Role – An example role is key which will then drive the sorting order of the code that is being developed.
7. Comments – Similar to the dataset level comments, the variable comments can be used as a status field for project managers to track the development of derived variables. This can document if the derivation is either in draft, development, testing or final production mode.

STEP 3: The dataset and variable level documentation is established during design and early stages of the study. This document is published to the entire group and can reside on a shared network drive for users to update. Project managers can use this tool to review the development and track updates and changes to the data and variable attributes. This document is a useful project management tool to track the status of the dataset all the way to the end when it can then be used for domain documentation as a Define.PDF document.

DATA STANDARDS

The domain documentation describes what data and variable attributes are going to be used for your study. This can reveal deviations from internal standards or industry standards such as CDISC. There are two general approaches towards achieving data standards. If you are starting from scratch, it makes sense to use a suggested standard such as CDISC. In this case, the effort will be in ensuring that new data created adheres to this standard. A second approach is when you already have existing data that is structured very different from CDISC standards. In this case, the task is to make sure that all existing data structures follow an internal standard. Any new data created would then need to adhere to this new standard. It is more common for the second scenario to occur.



Establishing data standards and applying the standards across all studies and projects can be resource intensive. It is reasonable to ask the question whether it is worth all the effort. One of the key benefits is that the programs associated with this data become more portable. They can be moved from one study to the next with minor modifications. Not only are the programs more portable, the programmer and statistician working on one study can understand a new study with the same structure relatively quickly compared to learning a new set of programs, macros and data structures. The productivity gain is sometimes difficult to measure but, in the long run, it will outweigh the efforts invested in standardizing.

You can follow a few simple steps with the use of the domain documentation as a target for your study.

STEP 1: Verify at the dataset level if the dataset names used in your study adhere to existing internal or CDISC standards. During design stage, you would just compare the DEFINE.RTF (or DEFINE.XML) against your existing standards.

STEP 2: Verify the variable names and related attributes to see if they match the existing internal standards or CDISC standards. Similar to step 1, this can be a visual inspection during the design stage of your study.

STEP 3: Once your programs are developed and you have physical SAS datasets created with these variables, it is important to reconcile these data sets and related attributes against the standards. Even with the best design, the programming and creation of these datasets always leads to deviations.

You can do this with visual inspections of the PROC CONTENTS results. You can also perform this task in an automated fashion using tools such as %cdisc and %diffetest. This automates the process of performing the comparison algorithmically to catch the most subtle differences and deviations which you may miss through visual inspection. These tools are available as part of the CDISC Builder tools available from MXI.

Data standards are an important step towards becoming more effective and efficient during your analysis programming and reporting. Accomplishing standards leads to portability of data between studies, while also increasing the mobility of team members between projects. This can increase consistency of data for accuracy and decrease validation efforts. Depending on the history and environment of your data, it can be beneficial to use CDISC or develop your own internal data standards. In either case, it is important to stage the release of standards within your group at logical time points to avoid dramatic changes to existing standard software. Once you have your data standards applied, the maintenance can be automated to a degree. You can use the domain documentation as documentation and tracking mechanism to manage these standards.

DATA INTEGRITY AND REVIEW

You can use the DEFINE.PDF/DEINF.XML files that are created for electronic submission to review your own data. This is usually performed by an independent reviewer outside of the development team. The fresh perspective from the reviewer creates a redundancy that ensures the accuracy and integrity of your data. This would allow you to catch discrepancies that may otherwise be captured during a review from regulatory agencies. There are steps which you can perform to ensure that your domain documentation is accurate and that the data which it is describing is accurate.

STEP 1: Verify that any hyperlinks such as the one to external transport (XPT) files link to the right files. This is to ensure that the domain document itself has accurate hyperlinks.

STEP 2: At the top list for datasets, verify the key fields. Ensure that the following criteria are met:

1. The key field exists and is listed first in the list of variables.
2. The dataset is sorted by the key fields.

STEP 3: Verify all decoded formats. Verify that the values of the decodes match what was defined in the analysis plan or original case report form. Review the data to see if there are any values that do not meet the formatted codes and therefore was not properly de-coded.

STEP 4: All derived variables need to be verified. You can choose to do some or all of the following recommended verification tasks to ensure the integrity of the derived variables:

Code Review	Systematic review of program code pertaining to the derivation according to a predetermined checklist of verification criteria.
Code Testing	Perform testing on SAS programs pertaining to the derivation supplying valid and invalid inputs and verify expected output.
Log Evaluation	Evaluate the SAS log for error, warning and other unexpected messages.
Output Review	Visual or programmatic review of report outputs related to the derivation as compared to expected results.
Data Review	Review attributes and contents of output data for accuracy and integrity.
Duplicate Programming	Independent programming to produce the same derivation and output for

comparison.

For the code review, an example list of verification criteria is shown here:

```
SAS PROGRAM
___ Program header complete and up-to-
    date.
___ Program commented clearly and
    adequately.
___ Macro variable definitions and
    usage are correct.
___ Data 'hardcode' corrections clearly
    commented with references in both
    the header and program.
___ Sorting orders correct.
___ Variables have been defined in the
    input datasets.
___ Macros used in more than one
    program are in the tools dir.

SAS LOG
___ Error check program clear.
___ Warning and error messages and
    warnings eliminated.
___ Un-initialized variables and merge
    notes eliminated.
___ Number of observations in each
    dataset as expected.
___ Checked all debugging messages
    (e.g. PUT statements).

SAS OUTPUT
___ All results have be validated
    either by outside methods (e.g.,
    proc freq or means, other) or
    against the CRT file (listing or
    SAS viewer).
___ Titles, footnotes and labels
    verified for accuracy and
    completeness.
___ Spelling is correct.  Checked for
    typos.
___ Labels for columns and rows are
    correct.
```

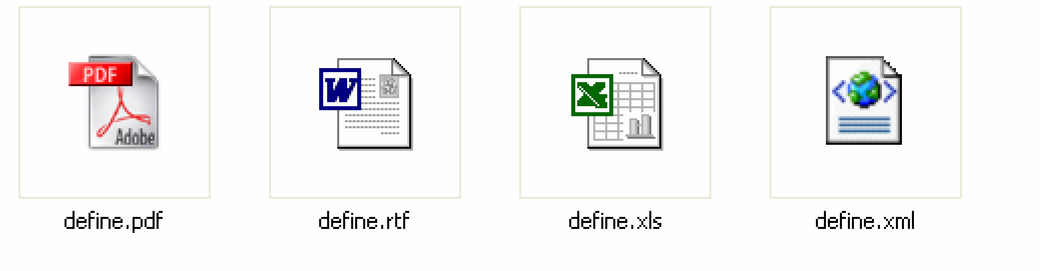
The checklist will vary with each group since there may be some departmental standards that are not universally applied. In general, however, the essence of the list is the same. The goal of the checklist is to ensure that independent reviewers do all the verification tasks required. This also ensures that the discrepancies found corresponding to the checklist are easily documented and understood across different programs pertaining to derivations.

There are many tasks performed in the process of verifying and validating SAS programs to ensure the quality of your data. Many of these tasks are overlooked for their significance in maintaining accuracy and integrity of the program logic and output which it produces. The repetitive aspect of these tasks gives them a bad reputation of being unglamorized grunt work that must be done to meet departmental SOPs. However, it is an essential step which can be performed and directed by what is documented in the domain documentation.

FILE FORMATS

The domain documentation originally was specified to be a DEFINE.PDF file. The PDF format is good in that it is not intended to be edited and can be viewed both on screen and on printed paper on many computing platforms. This is a good file format for the final electronic submission but if it is used for other purposes, other formats may be more suitable. PDF does have limitations in that it is not extensible. You cannot add extra information. For example, if you wanted to store information about the user name and date and time as to when they have last updated a particular variable, you cannot easily do this within the current DEFINE.PDF. However, XML file format is extensible which allows you to add more information. The new standard therefore is calling for the documentation to be stored in a more vendor neutral, universal and extensible structure of DEFINE.XML.

If you are to use the domain documentation for project management, the information can be stored in either an Excel spreadsheet or a Word document. For electronic submissions, the XML or DEFIND.PDF is a better choice. There may be slight variations but the main core information is the same for these files.



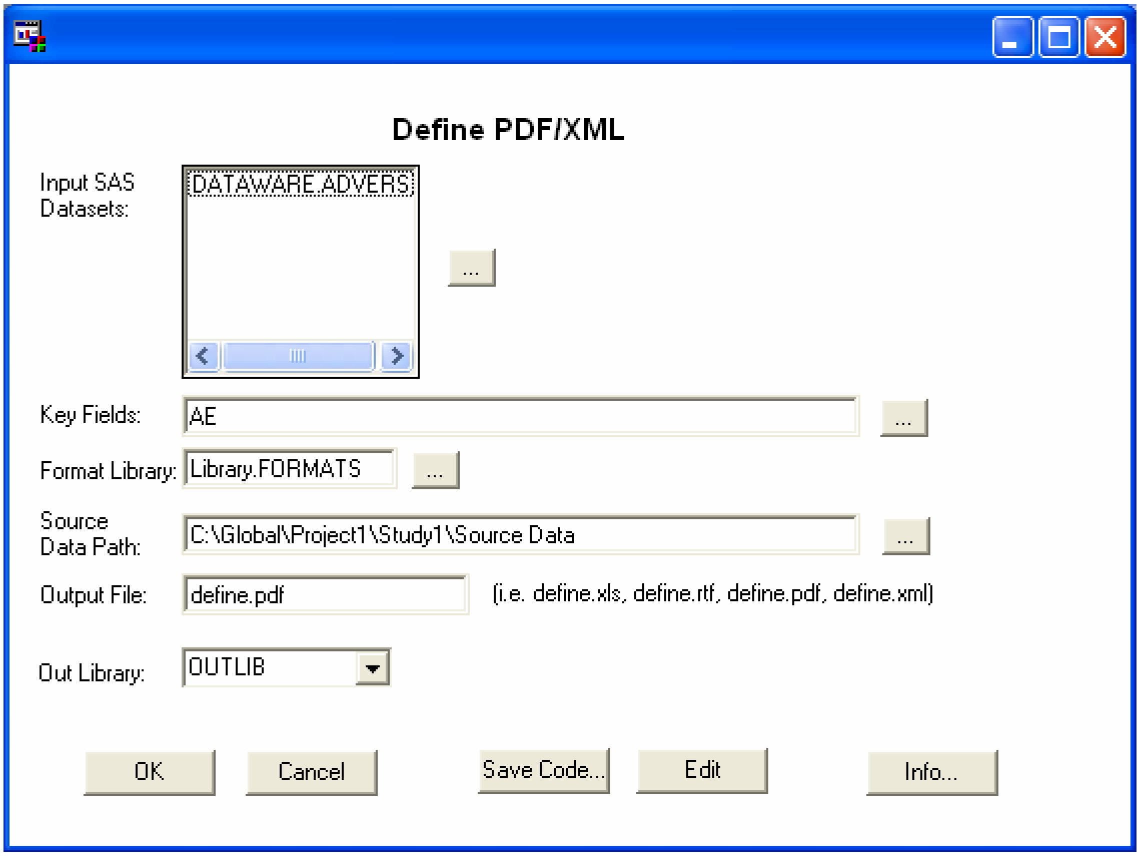
Since the content of the data is similar, file format becomes less significant. The file can be converted from one format to the other while maintaining all the same information. There are tools that will make this a transparent process. The goal is to make use of the information stored within the domain documentation and not be restricted by being forced to use one particular file format.

AUTOMATED TOOLS

The best way to ensure the accuracy of your domain documentation is to have it created or refreshed with the data that you are about to submit. That means every dataset and related variable attributes has to be captured and reconciled with what is in the DEFINE.PDF/DEFINE.XML file. This can be a very resource intensive task if you were to have to perform this manually. Cutting and pasting information can help but manually creating these files can lead to mistakes. Automating these tasks where possible is what the Definedoc™ provides for you. This section describes a sample scenario where the tools are used to automate these tasks.

CAPTURING METADATA

If you want to refresh or update the domain documentation, you would invoke the Definedoc application. It prompts you with all the attributes needed to automate the capture of information needed to generate the domain documentation.



Define PDF/XML

Input SAS Datasets: DATAWARE.ADVERS

Key Fields: AE

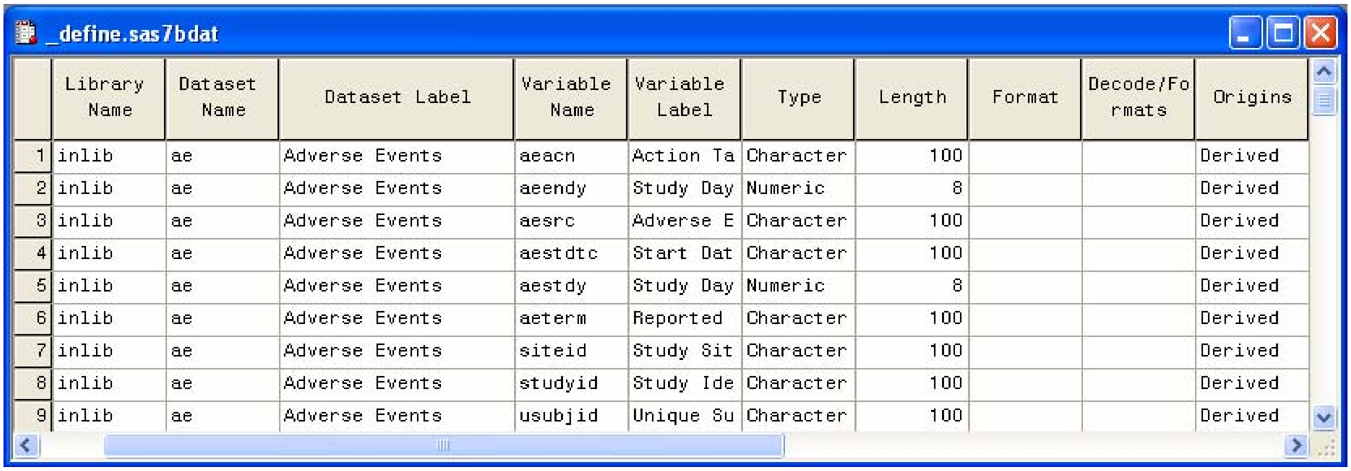
Format Library: Library.FORMATS

Source Data Path: C:\Global\Project1\Study1\Source Data

Output File: define.pdf (i.e. define.xls, define.rtf, define.pdf, define.xml)

Out Library: OUTLIB

This dialog box initiates the metadata capture of the specified dataset. The user can select the input data which will be captured. Each time a user selects a new data source, it would append or update to an internal dataset named _DEFINE which is a SAS dataset version of the DEFINE.PDF. This is stored in the output library that is specified. A sample of the dataset would look like:



_define.sas7bdat

	Library Name	Dataset Name	Dataset Label	Variable Name	Variable Label	Type	Length	Format	Decode/Formats	Origins
1	inlib	ae	Adverse Events	aeacn	Action Ta	Character	100			Derived
2	inlib	ae	Adverse Events	aeendy	Study Day	Numeric	8			Derived
3	inlib	ae	Adverse Events	aesrc	Adverse E	Character	100			Derived
4	inlib	ae	Adverse Events	aestdtc	Start Dat	Character	100			Derived
5	inlib	ae	Adverse Events	aestdy	Study Day	Numeric	8			Derived
6	inlib	ae	Adverse Events	aeterm	Reported	Character	100			Derived
7	inlib	ae	Adverse Events	siteid	Study Sit	Character	100			Derived
8	inlib	ae	Adverse Events	studyid	Study Ide	Character	100			Derived
9	inlib	ae	Adverse Events	usubjid	Unique Su	Character	100			Derived

This single _DEFINE dataset can then be used to generate any of the formats that you wish including: XLS, RTF, PDF, and XML.

BATCH CODE

The graphical user interface is a friendly tool for you to get started when capturing the metadata. However, if you are to perform these tasks repeatedly for tens and sometimes hundreds of datasets, it is useful to have a batch equivalent. If you were to click on the “Save Code…” button, Definedoc will generate the code to then perform the same task. The code equivalent of the dialog box is a SAS macro which has the following syntax.

definepdf

Generate a document containing metadata of information use to generate DEFINE.PDF

```
%definepdf (data = SAS data for documentation,
            keys = key fields,
            fmtlib = format library,
            source = source data path,
            outlib = output libname,
            output = output spreadsheet);
```

An example call to the macro would therefore look like:

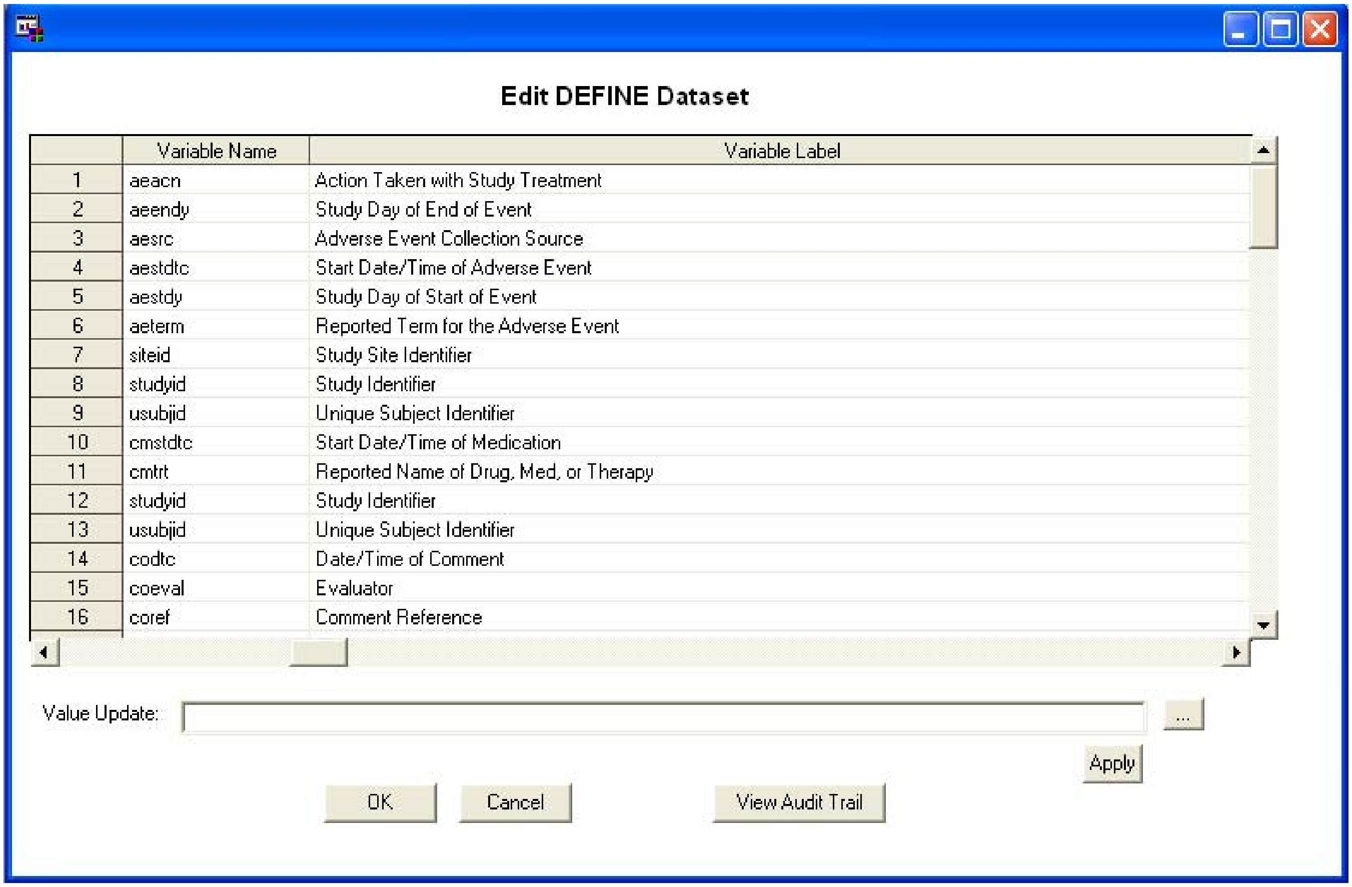
```
libname dataware 'c:\mypath\to\data';

%definepdf(data =dataware,
   source=C:\path\to\source\data,
   fmtlib =library.formats,
   output =define.xls,
   keys =ptid);
```

You can use the graphical user interface to produce template code but once you are familiar with the syntax, a script file can be generated to capture the metadata of all your data in an efficient batch approach.

EDITING INFORMATION

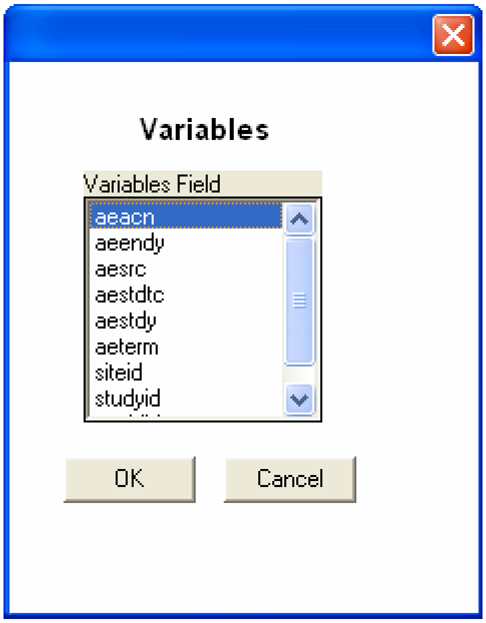
The automated capture of the information performs functions such as PROC CONTENTS to capture the metadata. It performs an educated guess as to the origin of the variable by searching through the source datasets. If it finds a match, it defaults the value to source. Otherwise, it considers the variable derived. There are instances however when this estimate is not accurate. Definedoc allows you to edit the _DEFINE dataset with the following interface.



Edit DEFINE Dataset

	Variable Name	Variable Label
1	aeacn	Action Taken with Study Treatment
2	aeendy	Study Day of End of Event
3	aesrc	Adverse Event Collection Source
4	aestdtc	Start Date/Time of Adverse Event
5	aestdy	Study Day of Start of Event
6	aeterm	Reported Term for the Adverse Event
7	siteid	Study Site Identifier
8	studyid	Study Identifier
9	usubjid	Unique Subject Identifier
10	cmstdtc	Start Date/Time of Medication
11	cmtrt	Reported Name of Drug, Med, or Therapy
12	studyid	Study Identifier
13	usubjid	Unique Subject Identifier
14	codtc	Date/Time of Comment
15	coeval	Evaluator
16	coref	Comment Reference

The information is presented in a spreadsheet like view which can be edited. Additionally, the user can click on the […] button. Depending on which field the user is on, a different dialog box is presented to present users with selection. For example, if the user was reviewing variable names, the following screen is shown:



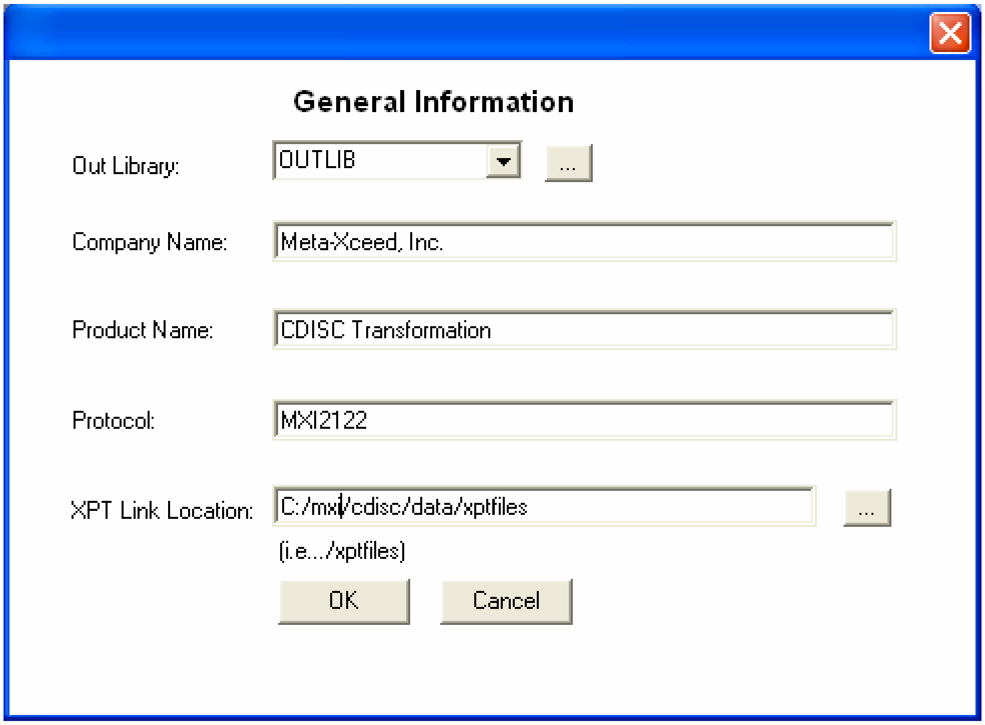
Variables

Variables Field

aeacn
aeendy
aesrc
aestdtc
aestdy
aeterm
siteid
studyid

The selected values are derived from the input dataset. The user can then select any value and apply to one or more of the rows within the spreadsheet view. This can help update information from the input dataset with greater accuracy.

Besides the metadata information that comes from PROC CONTENTS, there is other administrative information from the study. This information is captured through a separate information screen.



General Information

Out Library: OUTLIB

Company Name: Meta-Xceed, Inc.

Product Name: CDISC Transformation

Protocol: MXI2122

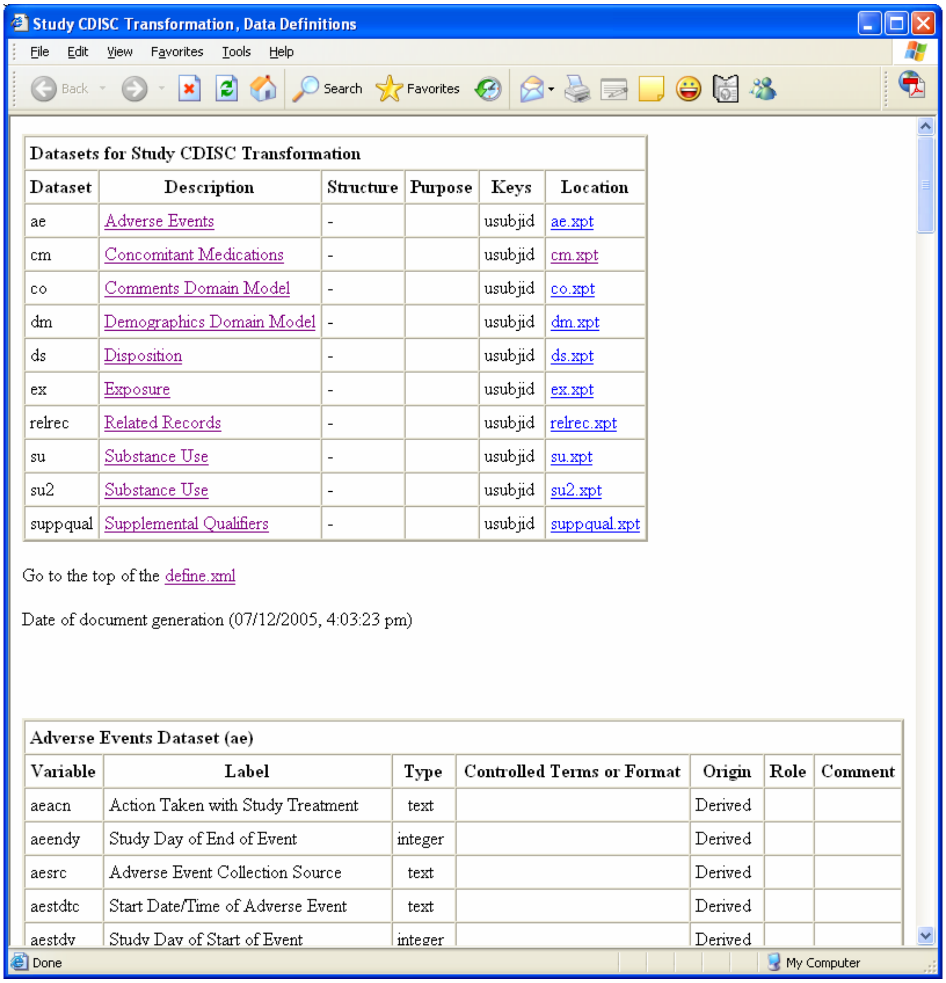
XPT Link Location: C:/mxi/cdisc/data/xptfiles

(i.e.../xptfiles)

This information is also stored in the _DEFINE dataset and is used to generate the final domain documentation. Every time you update the information, an audit trail of this information is captured in another dataset _DEFINEA. The audit trial records which user and at what time the edits were applied. This is essential in tracking down how a particular value was updated in event of a discrepant update.

GENERATING DEFINE.PDF/XML

The final step is to generate the DEFINE.PDF file. The Definedoc allows you to create it in any format you choose. This includes Define.PDF, Define.XML, Define.RTF and Define.XLS. You can create this by going through the user interface or re-running the batch program that calls the %definepdf macro. The PDF, RTF and XLS files are easily reviewed. Acrobat Viewer, Microsoft Word and Excel will open up these formats so you can see them. However, the XML files are intended to be viewed by a machine and do not have a visual component to them by default. The Definedoc tool does also generate an accompanying XSL Stylesheet for the XML file so you can view it directly in a web browser.


Study CDISC Transformation, Data Definitions
File Edit View Favorites Tools Help
Back
Search
Favorites
Done
My Computer

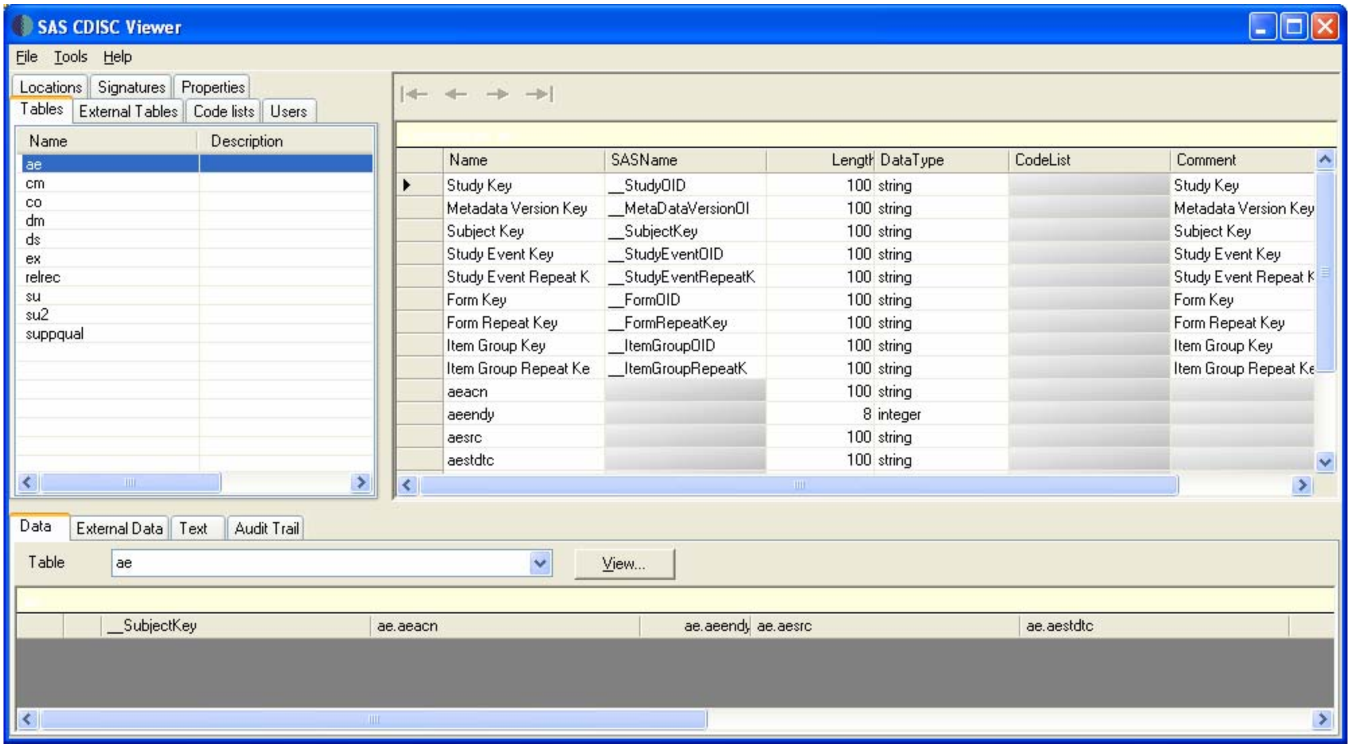
Datasets for Study CDISC Transformation					
Dataset	Description	Structure	Purpose	Keys	Location
ae	Adverse Events	-		usubjid	ae.xpt
cm	Concomitant Medications	-		usubjid	cm.xpt
co	Comments Domain Model	-		usubjid	co.xpt
dm	Demographics Domain Model	-		usubjid	dm.xpt
ds	Disposition	-		usubjid	ds.xpt
ex	Exposure	-		usubjid	ex.xpt
relrec	Related Records	-		usubjid	relrec.xpt
su	Substance Use	-		usubjid	su.xpt
su2	Substance Use	-		usubjid	su2.xpt
suppqual	Supplemental Qualifiers	-		usubjid	suppqual.xpt

Go to the top of the define.xml

Date of document generation (07/12/2005, 4:03:23 pm)

Adverse Events Dataset (ae)						
Variable	Label	Type	Controlled Terms or Format	Origin	Role	Comment
aeacn	Action Taken with Study Treatment	text		Derived		
aeendy	Study Day of End of Event	integer		Derived		
aesrc	Adverse Event Collection Source	text		Derived		
aestdtc	Start Date/Time of Adverse Event	text		Derived		
aestdy	Study Day of Start of Event	integer		Derived		

This gives you functionality that would not be available if you were to create the XML file by itself. The format of the XML file is also generated in a standard manner so if you were to use the SAS CDISC Viewer, you can open the same XML file and view it.



Name	Description
ae	
cm	
co	
dm	
ds	
ex	
relrec	
su	
su2	
suppqual	

Name	SASName	Length	DataType	CodeList	Comment
Study Key	__StudyOID	100	string		Study Key
Metadata Version Key	__MetaDataVersionOI	100	string		Metadata Version Key
Subject Key	__SubjectKey	100	string		Subject Key
Study Event Key	__StudyEventOID	100	string		Study Event Key
Study Event Repeat K	__StudyEventRepeatK	100	string		Study Event Repeat K
Form Key	__FormOID	100	string		Form Key
Form Repeat Key	__FormRepeatKey	100	string		Form Repeat Key
Item Group Key	__ItemGroupOID	100	string		Item Group Key
Item Group Repeat Ke	__ItemGroupRepeatK	100	string		Item Group Repeat Ke
aeacn		100	string		
aeendy		8	integer		
aesrc		100	string		
aestdtc		100	string		

Table: ae

__SubjectKey	ae.aeacn	ae.aeendy	ae.aesrc	ae.aestdtc

Automating the task of generating the domain documentation can provide many features and functionality that you would not otherwise have if you were to generate the files manually. It also automates the capture of the metadata with complete audit trail for you to be accurate and compliant with regulatory requirements.

CONCLUSION

The domain documentation is intended to guide reviewers to the information that is needed to expedite approval process. This is very important in that time wasted during approval can lead to great loss in profits for pharmaceutical companies. Besides the use of this tool for review, it can also be used internally among SAS programmers and biostatisticians during the analysis of the study. This provides project management, data standards and integrity. By using the document internally, it also ensures that the final product is tested and does not have bad hyperlinks or discrepant metadata. The accuracy of this document depends on how closely it reflects the data that is to be submitted. Since the data may change and refreshing and reconciling the documentation occurs frequently, automation is important. If left up to manual updates, this can be a resource intensive task that leads to many mistakes. The Definedoc utility described in this paper automates the mundane tasks but also adds features of different file formats to provide more flexibility and brings more value to the domain documentation.

REFERENCES

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.

Definedoc and all other **MXI** (Meta-Xceed, Inc.) product names are registered trademarks of Meta-Xceed, Inc. in the USA.

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

ABOUT THE AUTHOR

Sy Truong is President of MXI (Meta-Xceed, Inc.) They may be contacted at:

Sy Truong

48501 Warm Springs Blvd. Ste 117

Fremont, CA 94539

(510) 226-1209

sy.truong@meta-x.com