

SDTM, ADaM and define.xml with OpenCDISC®

Matt Becker and Angela Ringelberg, PharmaNet, Cary, NC

ABSTRACT

Standards are an ongoing focus of the health care and life science industry. Common terms you will see and hear during industry conferences include: "SDTM," "ADaM," "ODM," "LAB," "SEND," and "define(.xml/.pdf)." What do these terms mean? How do we create and validate these standards before submission to a client OR the FDA? Is there an easier way to ensure compliance? As a user, many of us have spent hours reading the SDTM/ADaM standards and implementation guides to generate "compliant" SAS data sets for our clinical studies. We have spent countless hours having another user QC our data structures...but is there an easier way?

Each of these terms and subsequent standards encompass a set of metadata. Metadata is "data about data." For example, a Proc Contents of a data set produces a list of variables, variable type, variable length, variable format, etc. Each of these tidbits of information is metadata about that individual data set. SDTM defines standard metadata for "domains." Similarly, ADaM defines standard metadata for analysis data sets. If that metadata is standard it should be feasible to construct an application that will check your data's metadata against that standard's metadata.

OpenCDISC is an open source community which is focusing on creating frameworks and tools for the implementation and advancement of CDISC Standards. OpenCDISC has created a CDISC Validator which will eliminate the need for individuals to develop their own custom processes in order to ensure that their CDISC models are compliant with CDISC standards. By taking common validation rules, OpenCDISC has developed an open-source tool which is freely available and of commercial-quality to ensure data compliance with CDISC models such as SDTM, ADaM, and Define.xml. The validation rules for each standard have been pooled into a CDISC Validation Rules Repository, providing users with a central listing. The listing is easy to use, contribute to, improve on and continue development.

In this Hands-On Workshop (HOW) we are going to briefly describe a few of the key terms (SDTM, ADaM, define) and investigate the use of OpenCDISC Validator to perform the following tasks:

- Validate SDTM 3.1.1 SAS data sets
- Validate SDTM 3.1.2 SAS data sets
- Validate ADaM 1.0 SAS data sets
- Generate define.xml

INTRODUCTION

Before going too much further with the discussion of OpenCDISC, a quick overview of CDISC (Clinical Data Interchange Standards Consortium) is merited. Over the past few years CDISC has become common terminology in our workplace and we have started to use CDISC standards in our work more and more. The CDISC standards provide data consistency across the spectrum and this standardization has helped streamline drug development.

In this paper, we are going to concentrate on the SDTM, ADaM, and define.xml CDISC standards. SDTM (Study Data Tabulation Models) is the content standard of case report form data tabulations from clinical research studies. ADaM (Analysis Data Model) is the content standard of analysis datasets. Define.xml (Case Report Tabulation Data Definition Specification (CRTDDS)) is an XML-based content and format standard which contains the specifications for data definitions for CDISC SDTM datasets.

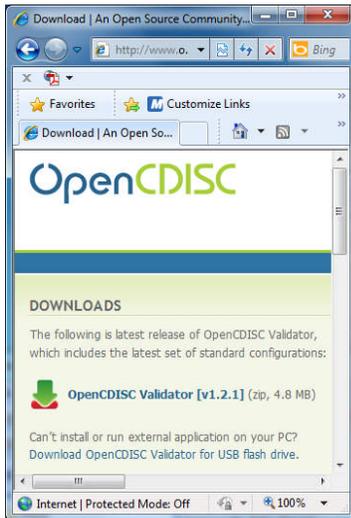
When we create SDTM files, ADaM files, and/or define.xml, we must make sure that they are compliant with CDISC standards. We must check our work. How is this done? Usually, by double programming (at least in the case of SDTM and ADaM files). The re-creation of the files by an independent programmer and comparing the two sets of results. This is no simple task. It requires a lot of time and a lot of reconciliation between the production programming and the validation programming in order to make sure there is compliance with the CDISC standards. And, once this process is complete, how can we guarantee 100% compliance? The individualized validation process

for compliance with the CDISC standards is not a standardized task; each of us develops our own ways of validating our files.

Here is where OpenCDISC comes into the picture. OpenCDISC has created a CDISC Validator which will eliminate the need for individuals to develop their own custom processes. The OpenCDISC Validator ensures that your CDISC models are compliant with CDISC standards. OpenCDISC has taken common validation rules and pooled them into a CDISC Validation Rules Repository providing users with a central listing. The Validator is free and easy to use.

USING THE OPENCDISC VALIDATOR

The validator requires Java Runtime Environment (JRE) version 1.5 or higher and 2GB system RAM. Download the OpenCDISC validator from <http://www.opencdisc.org>, click on the OpenCDISC Validator [v1.2.1] link and unzip to your chosen directory. Detailed installation directions are provided on the website.

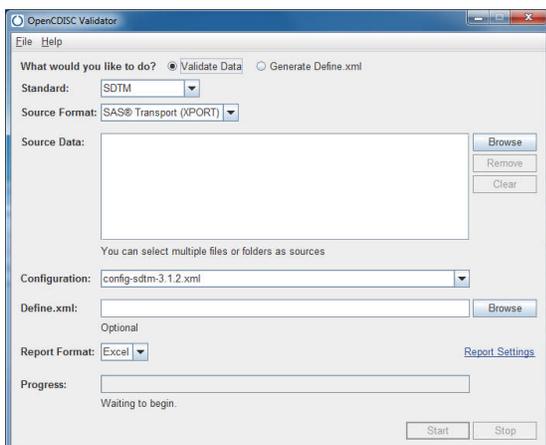


Once the validator has been downloaded and unzipped, it is ready to use.

VALIDATING SDTM FILES

Step 1: Open the 'opencdisc-validator' folder.

Step 2: Double click on the 'client.bat' file. This will bring up the OpenCDISC Validator window:

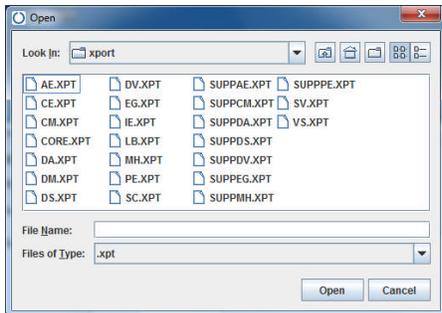


Step 3: For the question "What would you like to do?" select 'Validate Data'.

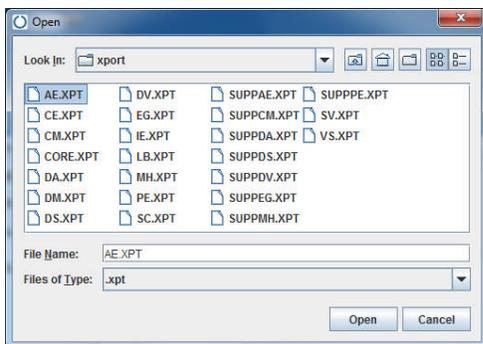
Step 4: Choose the Standard (the default is SDTM). For this example, we chose SDTM.

Step 5: Choose the Format (the default is XPORT). Note that the SDTM files must be in SAS® Transport (XPORT) or a delimited file. The validator cannot process regular SAS datasets.

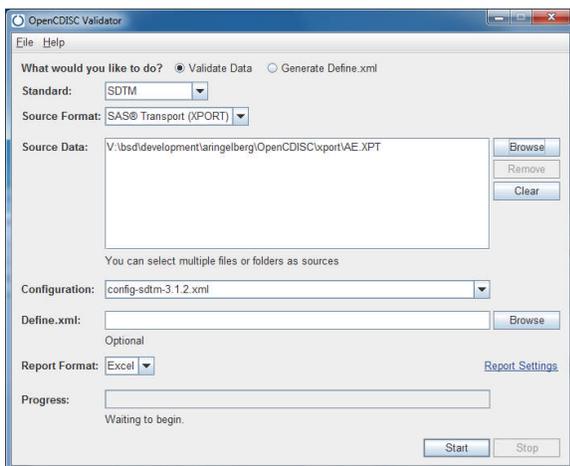
Step 6: Choose the source data by clicking on the Browse button on the right hand side. The following window will appear:



Step 7: Highlight the SDTM file or files you want to check.



Step 8: Click Open. The OpenCDISC validator window will appear with the files or files you have selected in the Source Data field.

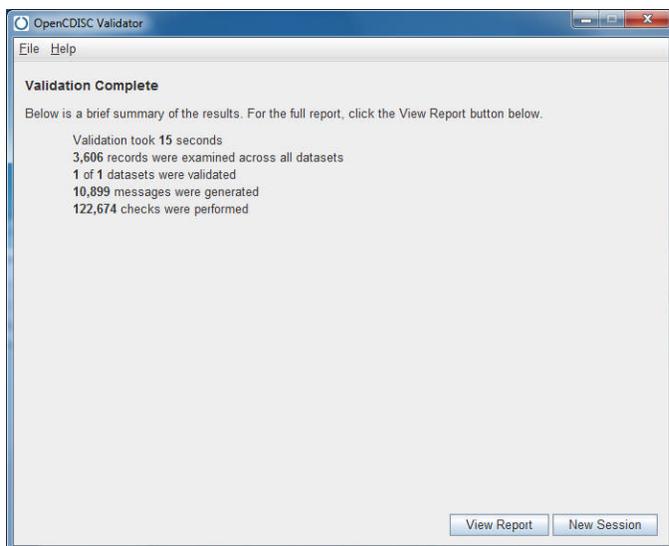


Step 9: Choose the configuration. The default for SDTM files is sdtm-3.1.2.xml.

Step 10: Choose the report format. The default is Excel.

Step 11: In this example, we are now ready to start the validation of the AE SDTM file. Click the start button.

When the validation is complete, you will receive an information window providing you with how long it took for the validator to run, the number of records read, the number of datasets validated, the number of messages generated and the number of checks performed:



Step 12: At this point, you may choose to view the report or start a new session. Choose 'View Report'. The report will consist of 4 tabs within the Excel document: Dataset Summary, Issue Summary, Details and Rules.

The **Dataset Summary** tab provides a brief overview of what was encountered by the validator.

Name	Label	Class	Source	Records	Errors	Warnings	Infos
AE	Adverse Events	Events	AE.XPT	3606	79	10816	3
Total				3606	79	10816	3

Name	Label	Class	Reason	Errors	Warnings	Infos
Total				0	0	0

Grand Total				3606	79	10816	3
-------------	--	--	--	------	----	-------	---

The report tells us that there are 3606 records in the AE file that 79 errors were encountered, and 10,816 Warnings were reported.

The **Issues Summary** tab provides break down of the type of rules that have issues and how many have been reported.

OpenCDISC Validator Report		
Configuration: C:\Users\aringelberg\Desktop\OpenCDISC\opencdisc-validator\config\config-sdtm-3.1.2.xml		
Define.xml: Not provided		
Generated: 2011-03-10T16:51:19		
Issue Summary		
Rule ID	Message	Found
Error		
SD0002	Null value in variable marked as Required	77
SD0013	Begin day must be less than or equal to end day	2
SD1020	Missing DM dataset	1
Total		80
Warning		
CT0001	Value for AEACN not found in ACN controlled terminology codelist	3586
CT0027	Value for AEOUT not found in OUT controlled terminology codelist	3197
CT0037	Value for AEBODSYS not found in SOC controlled terminology codelist	3529
SD0009	AE is Serious but no qualifiers set to 'Y'	499
SD0031	Start date expected when end date provided	5
Total		10816
Information		
SKIP_SD0064	The source data for DM is missing and cannot be used for cross-dataset validation	1
SKIP_SD0080	The source data for DS is missing and cannot be used for cross-dataset validation	1
SKIP_SD1005	The source data for DM is missing and cannot be used for cross-dataset validation	1
Total		3

The **Details** tab provides us just that...the details. Each error or warning message has been expanded. Below is a small sample of what this AE file's issues log generated:

	A	B	C	D	E	F	G	H
1	Name	Record	Variables	Values	Rule ID	Message	Category	Type
64	AE	19	AEOUT	RECOVERED	CT0027	Value for AEOUT not found in OUT controlled terminology codelist	Terminology	Warning
65	AE	19	AEBODSYS	Neoplasms benign, malignant and unspecified (incl cysts and polyps)	CT0037	Value for AEBODSYS not found in SOC controlled terminology codelist	Terminology	Warning
66	AE	20	AEACN	NONE	CT0001	Value for AEACN not found in ACN controlled terminology codelist	Terminology	Warning
67	AE	20	AEOUT	RECOVERED	CT0027	Value for AEOUT not found in OUT controlled terminology codelist	Terminology	Warning
68	AE	20	AEDECOD	null	SD0002	Null value in variable marked as Required	Presence	Error
69	AE	21	AEACN	NONE	CT0001	Value for AEACN not found in ACN controlled terminology codelist	Terminology	Warning
70	AE	21	AEOUT	ADVERSE EVENT STILL PRESENT	CT0027	Value for AEOUT not found in OUT controlled terminology codelist	Terminology	Warning

And finally, the **Rules** tab shows us the standards. Here is a small sample:

Rule ID	Message	Description	Category	Type
CT0001	Value for --ACN not found in ACN controlled terminology codelist	Variable values should be populated with terms found in 'Action Taken with Study Treatment' (C66767) CDISC controlled terminology codelist	Terminology	Warning
CT0002	Value for AESEV not found in AESEV controlled terminology codelist	Variable values should be populated with terms found in 'Severity/Intensity Scale for Adverse Events' (C66769) CDISC controlled terminology codelist	Terminology	Warning
CT0003	Value for TSVAL not found in AGESPAN controlled terminology codelist	Variable values should be populated with terms found in 'Age Span' (C66780) CDISC controlled terminology codelist	Terminology	Warning
CT0004	Value for AGEU not found in AGEU controlled terminology codelist	Variable values should be populated with terms found in 'Age Unit' (C66781) CDISC controlled terminology codelist	Terminology	Error
CT0005	Value for TSVAL not found in AGEU controlled terminology codelist	Variable values should be populated with terms found in 'Age Unit' (C66781) CDISC controlled terminology codelist	Terminology	Warning
CT0006	Value for COUNTRY not found in COUNTRY controlled terminology codelist	Variable values should be populated with terms found in 'Country' (C66786) CDISC controlled terminology codelist	Terminology	Warning
CT0007	Value for DATEST not found in DATEST controlled terminology codelist	Variable values should be populated with terms found in 'Drug Accountability Test Name' (C78731) CDISC controlled terminology codelist	Terminology	Warning
CT0008	Value for DATESTCD not found in DATESTCD controlled terminology codelist	Variable values should be populated with terms found in 'Drug Accountability Test Code' (C78732) CDISC controlled terminology codelist	Terminology	Warning
CT0009	Value for DOMAIN not found in DOMAIN controlled terminology codelist	Variable values should be populated with terms found in 'Domain Abbreviation' (C66734) CDISC controlled terminology codelist	Terminology	Warning
CT0010	Value for DSCAT not found in DSCAT controlled terminology codelist	Variable values should be populated with terms found in 'Category for Disposition Event' (C74558) CDISC controlled terminology codelist	Terminology	Warning
CT0011	Value for EGMETHOD not found in EGMETHOD controlled terminology codelist	Variable values should be populated with terms found in 'ECG Test Method' (C71151) CDISC controlled terminology codelist	Terminology	Warning
CT0012	Value for EGSTRESC not found in EGSTRESC controlled terminology codelist	Variable values should be populated with terms found in 'ECG Result' (C71150) CDISC controlled terminology codelist	Terminology	Warning

With the information obtained from the OpenCDISC validator, the user can now go back to the production SDTM file and correct any issues the validator has flagged.

VALIDATING ADAM FILES

Step 1: Open the 'opendisc-validator' folder.

Step 2: Double click on the 'client.bat' file. This will bring up the OpenCDISC Validator window.

Step 3: For the question 'What would you like to do?' select 'Validate Data'.

Step 4: Choose the Standard (the default is SDTM). For this example, we chose **ADaM**.

OpenCDISC Validator

File Help

What would you like to do? Validate Data Generate Define.xml

Standard: ADaM

Source Format: SAS® Transport (XPORT)

Source Data: Browse Remove Clear

You can select multiple files or folders as sources

Configuration: config-adam-1.0.xml

Define.xml: Browse

Optional

Report Format: Excel Report Settings

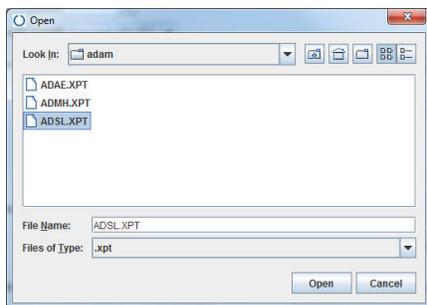
Progress: Waiting to begin.

Start Stop

Step 5: Choose the Format (the default is XPORT).

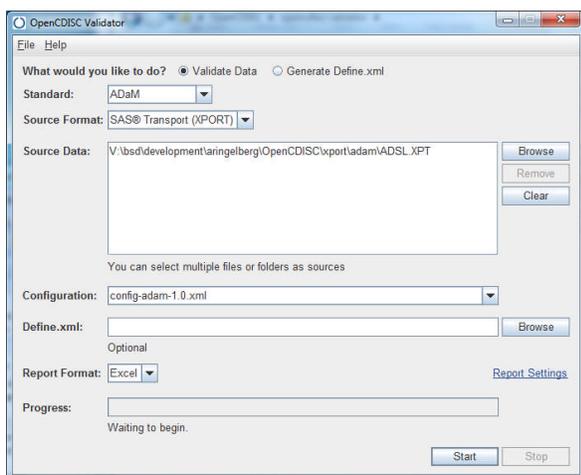
Step 6: Choose the source data by clicking on the Browse button.

Step 7: Highlight the ADaM file or files you want to check.



Step 8: Click Open. The OpenCDISC validator window will appear with the files or files you have selected in the Source Data field.

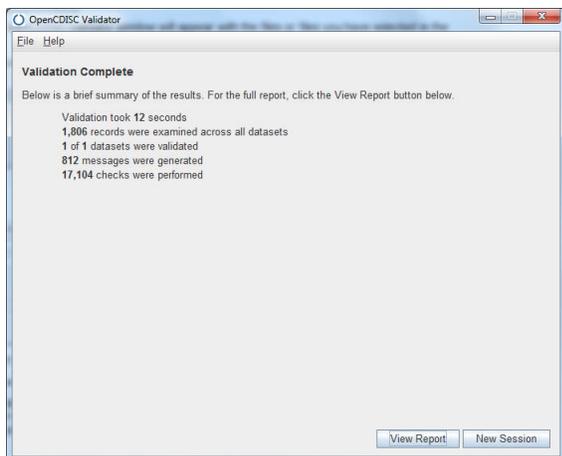
Step 9: Choose the configuration. Select 'config-adam-1.0.xml'. This is the default for ADaM files, and currently your only choice.



Step 10: Choose the report format. The default is Excel.

Step 11: In this example, we are now ready to start the validation of the ADSL ADaM file. Click the start button.

When the validation is complete, you will receive the information window providing you the same type of summary information as when we ran the STDM file validation.



Step 12: At this point, you may choose to view the report or start a new session. Choose 'View Report'. As with the SDTM validation, the ADaM validation provides a similar OpenCDISC Validator Report containing 4 tabs: Dataset Summary, Issue Summary, Details and Rules.

Dataset Summary:

OpenCDISC Validator Report						
Name	Label	Class	Source	Records	Errors	Warnings
ADSL	Subject-Level Analysis	Subject-Level Analysis	ADSL_XPT	1806	802	9
Total				1806	802	9
Unprocessed Sources						
Name	Label	Class	Reason	Errors	Warnings	Infos
Total				0	0	0
Grand Total				1806	802	9

Issue Summary:

Issue Summary		
Rule ID	Message	Found
Error		
AD0018	ADaM dataset variable label mismatch	2
AD0041	* DT does not have the ADaM required SAS Date format	5
AD0048	A population variable with a suffix of FL is not present in ADSL	1
AD0062	Variable of form TRTxxA does not have a 2-digit [01-99] number in the name for xx	1
AD0063	Variable of form TRTxxAN does not have a 2-digit [01-99] number in the name for xx	1
AD1001	Required AGEU is not present within dataset	1
AD1001	Required ARM is not present within dataset	1
AD1001	Required TRT01P is not present within dataset	1
AD1008	Null value in variable marked as Required	789
Total		802
Warning		
AD1002	Expected TR01EDT is not present within dataset	1
AD1002	Expected TR01SDT is not present within dataset	1
AD1002	Expected TRT01A is not present within dataset	1
AD1002	Expected TRTEDT is not present within dataset	1
AD1002	Expected TRTEDTM is not present within dataset	1
AD1002	Expected TRTSdT is not present within dataset	1
AD1002	Expected TRTSDTM is not present within dataset	1
AD1002	Expected TRTSEQA is not present within dataset	1
AD1002	Expected TRTSEQP is not present within dataset	1
Total		9
Information		
SKIP_AD0053	The source data for DM is missing and cannot be used for cross-dataset validation	1
Total		1

Details:

Name	Record	Variables	Values	Rule ID	Message	Category	Type
ADSL		VARIABLE	FASFL_PPROTFL, COM PLFL_ITFL_RANDFL_SA FFL_ENRFL	AD0048	A population variable with a suffix of FL is not present in ADSL	Presence	Error
ADSL		VARIABLE	AGEU	AD1001	Required AGEU is not present within dataset	Presence	Error
ADSL		VARIABLE	ARM	AD1001	Required ARM is not present within dataset	Presence	Error
ADSL		VARIABLE	TRT01P	AD1001	Required TRT01P is not present within dataset	Presence	Error
ADSL		VARIABLE	TRTEDTM	AD1002	Expected TRTEDTM is not present within dataset	Presence	Warning
ADSL		VARIABLE	TRT01A	AD1002	Expected TRT01A is not present within dataset	Presence	Warning
ADSL		VARIABLE	TR01EDT	AD1002	Expected TR01EDT is not present within dataset	Presence	Warning
ADSL		VARIABLE	TRTEDT	AD1002	Expected TRTEDT is not present within dataset	Presence	Warning
ADSL		VARIABLE	TRTSDTM	AD1002	Expected TRTSDTM is not present within dataset	Presence	Warning
ADSL		VARIABLE	TRTSEQP	AD1002	Expected TRTSEQP is not present within dataset	Presence	Warning
ADSL		VARIABLE	TR01SDT	AD1002	Expected TR01SDT is not present within dataset	Presence	Warning
ADSL		VARIABLE	TRTSEQA	AD1002	Expected TRTSEQA is not present within dataset	Presence	Warning
ADSL		VARIABLE	TRTSDT	AD1002	Expected TRTSDT is not present within dataset	Presence	Warning
ADSL		FORMAT, VARIABLE	DATE, RFSTDT	AD0041	* DT does not have the ADaM required SAS Date format	Format	Error
ADSL		FORMAT, VARIABLE	DATE, RFENDT	AD0041	* DT does not have the ADaM required SAS Date format	Format	Error
ADSL		VARIABLE, LABEL	AGE, Age (yr) at reference start date	AD0018	ADaM dataset variable label mismatch	Metadata	Error
ADSL		VARIABLE	TRTA	AD0062	Variable of form TRTxxA does not have a 2-digit [01-99] number in the name for xx	Format	Error
ADSL		VARIABLE	TRTAN	AD0063	Variable of form TRTxxAN does not have a 2-digit [01-99] number in the name for xx	Format	Error
ADSL		VARIABLE, LABEL	RANDDT, Randomization Date	AD0018	ADaM dataset variable label mismatch	Metadata	Error
ADSL		FORMAT, VARIABLE	DATE, RANDDT	AD0041	* DT does not have the ADaM required SAS Date format	Format	Error
ADSL		FORMAT, VARIABLE	DATE, TERMDT	AD0041	* DT does not have the ADaM required SAS Date format	Format	Error
ADSL		FORMAT, VARIABLE	DATE, DEATHDT	AD0041	* DT does not have the ADaM required SAS Date format	Format	Error
ADSL				SKIP_AD0053	The source data for DM is missing and cannot be used for cross-dataset validation	System	Information
ADSL	2	AGE	null	AD1008	Null value in variable marked as Required	Presence	Error
ADSL	4	AGE	null	AD1008	Null value in variable marked as Required	Presence	Error

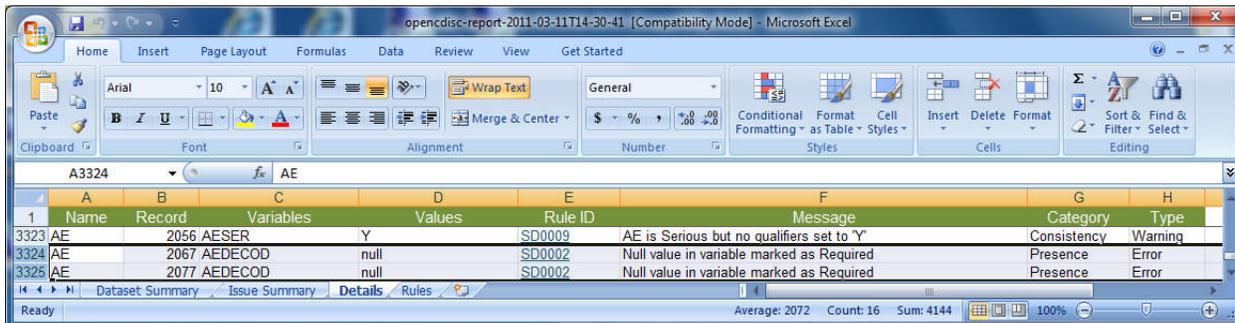
INTERPRETING THE OPENCDISC VALIDATOR OUTPUT

As we have seen, for both SDTM and ADaM files, the OpenCDISC Validator generates the same type of report. In the examples above, we requested that our report be put in Excel format. Within the Excel spreadsheet, the validator generated 4 information tabs, providing different levels of information on each tab. However, prior to viewing the report, the validator provides some feedback as to what it has encountered. With the SDTM AE file it reported that 3,606 records were read, 10,899 messages were generated, and 122,674 checks were performed. The number of messages generated is a bit overwhelming. So, let's take a closer look and see if we can figure out why there are so many issues with this file.

The Dataset Summary tab gives an overview of the issues encountered in the validation. It reported that encountered 79 errors and 10816 warnings. 79 errors are less daunting than the 10,899 messages. But we need to take a closer look. The Issue Summary tab provides a bit more detail. Breaking down the Error messages and Warning messages by type.

Rule ID	Message	Found
SD0002	Null value in variable marked as Required	77
SD0013	Begin day must be less than or equal to end day	2

Looking at the first error message (Rule ID: SD0002), we see that the validator found 77 occurrences for a variable that is NULL but is a REQUIRED variable. Now we are starting to zoom in on the problem. The Details tab does just that. Searching for Rule ID SD0002 we quickly determine that NULL values are for the variable AEDECOD.



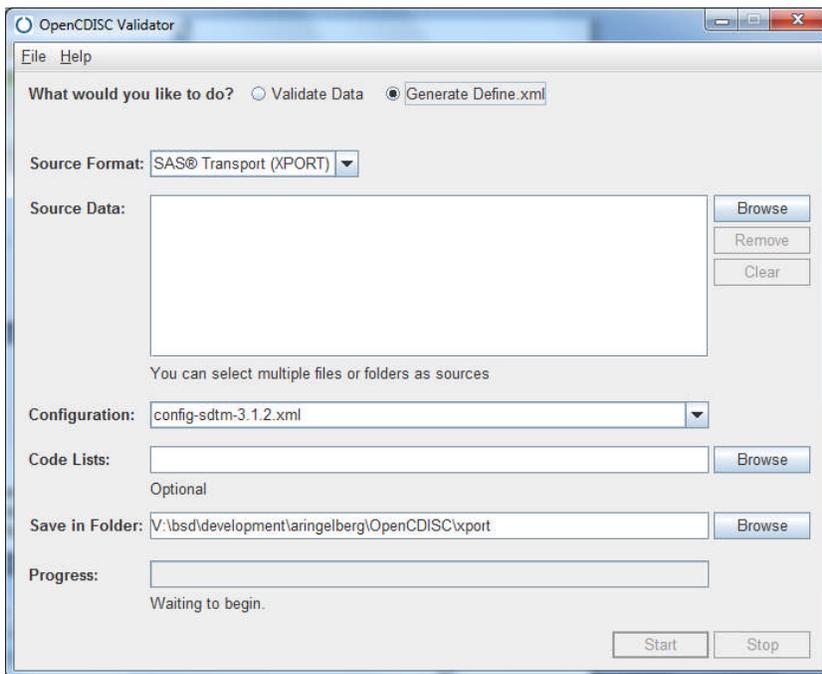
It looks like we have some adverse events that have not yet been coded. Knowing what the problem is, we can go back to the production dataset and make the correction.

The Rules tab is a for-your-information tab. It reports all of the validator rules by ID and provides the message, description, category and type for each rule.

CREATING DEFINE.XML

The OpenCDISC Validator can also generate DEFINE.XML. From the OpenCDISC Validator window perform the following steps.

Step 1: From the 'What would you like to do?' prompt, choose 'Generate Define.xml'.



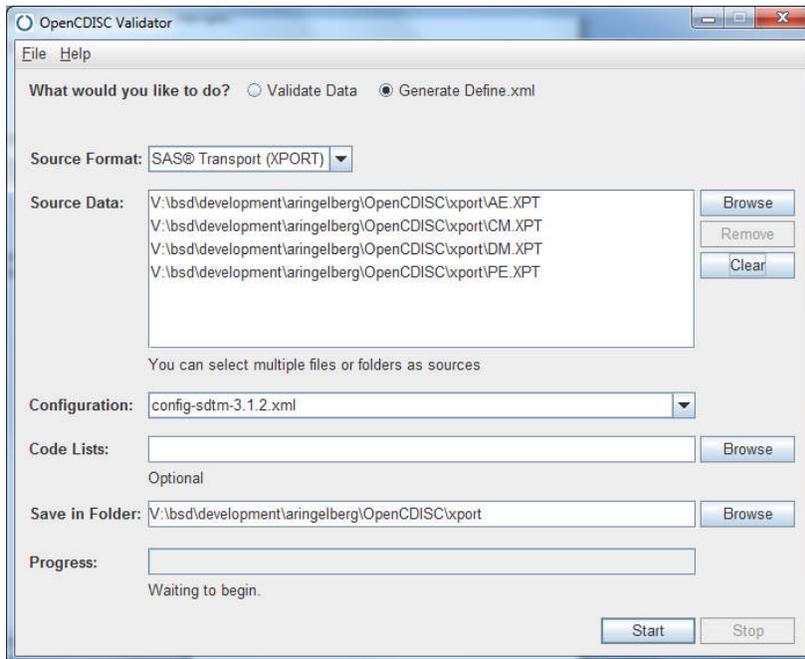
Step 2: Choose the Source Format. For this example we are using the default, XPORT.

Step 3: Choose the file(s) you want to create define.xml for using the Browse button.

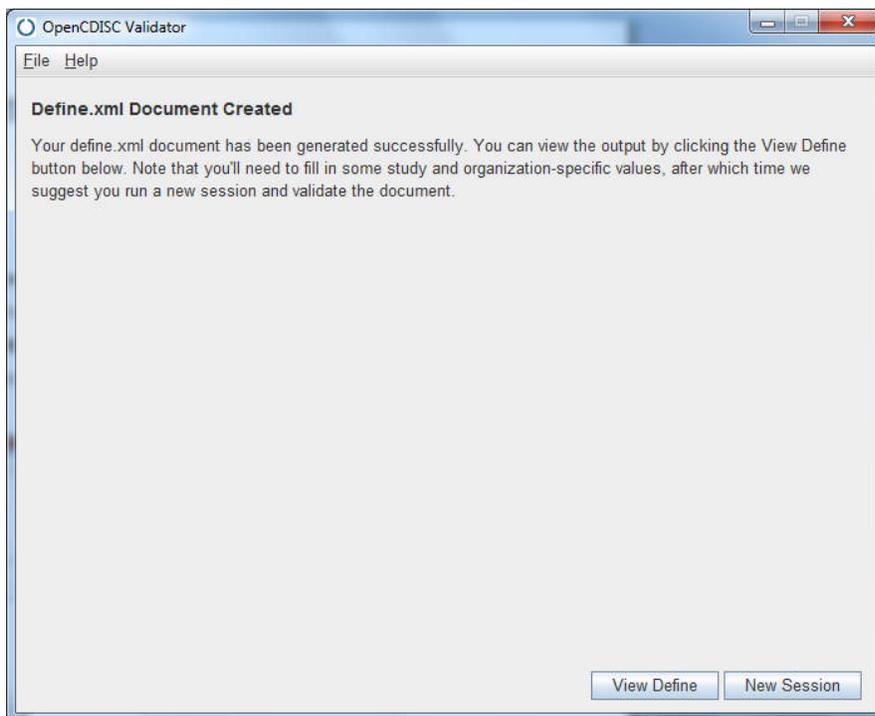
Step 4: Choose the Configuration. We are using 'config-sdtm-3.1.2.xml' for this example.

Step 5: Select the folder where you want to save the define.xml file.

Step 6: Click on Start.



When the define.xml file has been generated, the following information screen will appear:



Step 7: Choose 'View Define' to see the newly created define.xml file.

In this example, 4 SDTM files were selected: DM, CM, AE and PE. Below is some of the XML output. Note that some study and organization-specific values still need to be filled in.

Study, Data Definitions - Microsoft Internet Explorer provided by PharmaNET Technical Support

Go to the top of the [define.xml](#)

Date of document generation (2011-03-11T15:01:53)

Datasets for Study

Dataset	Description	Structure	Purpose	Keys	Location
DM	Demographics	Special Purpose - One record per subject	Tabulation		DM.XPT
CM	Concomitant Medications	Interventions - One record per medication intervention episode per subject	Tabulation		CM.XPT
AE	Adverse Events	Events - One record per event per subject	Tabulation		AE.XPT
PE	Physical Examinations	Findings - One record per event per subject	Tabulation		PE.XPT

Demographics Dataset (DM)

Variable	Label	Type	Controlled Terms or Format	Origin	Role	Comment
STUDYID	Study Identifier	text			Identifier	
DOMAIN	Domain Abbreviation	text			Identifier	
USUBJID	Unique Subject Identifier	text			Identifier	
SUBJID	Subject Identifier for the Study	text			Topic	
RFTDTC	Subject Reference Start Date/Time	datetime			Record Qualifier	
RFENDTC	Subject Reference End Date/Time	datetime			Record Qualifier	
SITEID	Study Site Identifier	text			Record Qualifier	
BRTHDTC	Date/Time of Birth	datetime			Record Qualifier	
AGE	Age	integer			Record Qualifier	
AGEU	Age Units	text			Variable Qualifier	
SEX	Sex	text			Record Qualifier	
RACE	Race	text			Record Qualifier	
ETHNIC	Ethnicity	text			Record Qualifier	
ARMCD	Planned Arm Code	text			Record Qualifier	
ARM	Description of Planned Arm	text			Synonym Qualifier	
COUNTRY	Country	text			Record Qualifier	

Concomitant Medications Dataset (CM)

Variable	Label	Type	Controlled Terms or Format	Origin	Role	Comment
STUDYID	Study Identifier	text			Identifier	
DOMAIN	Domain Abbreviation	text			Identifier	
USUBJID	Unique Subject Identifier	text			Identifier	
CMSEQ	Sequence Number	float			Identifier	
CMTRT	Reported Name of Drug, Med, or Therapy	text			Topic	
CMDECOD	Standardized Medication Name	text			Synonym Qualifier	
CMCAT	Category for Medication	text			Grouping Qualifier	
CMSTAT	Completion Status	text			Record Qualifier	
CMINDC	Indication	text			Record Qualifier	
CMCLAS	Medication Class	text			Variable Qualifier	
CMCLASCD	Medication Class Code	text			Variable Qualifier	
CMDOSTXT	Dose Description	text			Record Qualifier	
CMDOSE	Dose per Administration	float			Record Qualifier	
CMDOSU	Dose Units	text			Variable Qualifier	
CMDOSEFRM	Dose Form	text			Record Qualifier	
CMDOSEFRQ	Dosing Frequency per Interval	text			Variable Qualifier	
CMROUTE	Route of Administration	text			Variable Qualifier	
CMSTDTC	Start Date/Time of Medication	datetime			Timing	
CMENDTC	End Date/Time of Medication	datetime			Timing	
CMENRF	End Relative to Reference Period	text			Timing	

Adverse Events Dataset (AE)

Variable	Label	Type	Controlled Terms or Format	Origin	Role	Comment
STUDYID	Study Identifier	text			Identifier	
DOMAIN	Domain Abbreviation	text			Identifier	
USUBJID	Unique Subject Identifier	text			Identifier	
AESEQ	Sequence Number	float			Identifier	
AETEDTM	Reported Term for the Adverse Event	text			Topic	

CONCLUSION

The development of the open-source tool, the OpenCDISC Validator, is helping to ensure data compliance with CDISC models such as SDTM, ADaM and Define.xml. The tool is of commercial-quality, freely available and EASY to use. The tool eliminates the need for individualized QC that makes sure SDTM and ADaM files are CDISC compliant. The information generated by the validator is easy to interpret, allowing the user to make corrections in the affected SDTM or ADaM file - bringing that file up to CDISC standards. The validator also has the ability to create define.xml. It is as simple as bringing up the OpenCDISC Validator, choosing the option of 'Create define.xml' and instructing the validator which SDTM files need to be included. The OpenCDISC Validator saves a lot time and effort when it comes to the validation of SDTM and ADaM files and the creation of define.xml.

REFERENCES

[HTTP://WWW.CDISC.ORG](http://www.cdisc.org)

[HTTP://WWW.OPENCDISC.ORG](http://www.opencdisc.org)

CONTACT INFORMATION

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

Name: Matt Becker
Address: 1000 CentreGreen Way, Suite 300
City, State ZIP: Cary, NC 27513
Work Phone: 919-388-4031
E-mail: mbecker@pharmanet.com

Name: Angela Ringelberg
Address: 1000 CentreGreen Way, Suite 300
City, State ZIP: Cary, NC 27513
Work Phone: 609-951-6569
E-mail: aringelberg@pharmanet.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.