PharmaSUG2011 - Paper CD15

Validating define.xml: Tools, trials, and tribulations

Sandra VanPelt Nguyen, i3 Statprobe

ABSTRACT

The define.xml file should serve as a guide to the data which has been submitted for a clinical trial. For those who are unfamiliar with the trial and the data that was collected, documentation is critical to understanding, reviewing, and using that data. Without a proper roadmap, it is easy to get lost. Multiple types of quality control and validation steps are needed to ensure that the define.xml file is complete, accurate, functional, and consistent with the data it is supposed to represent and describe while additionally meeting the requirements and guidelines established in the Case Report Tabulation Data Definition Specification[1] (CRT-DDS, also known as define.xml).

Last year, CDISC released a white paper on XML schema validation for define.xml[2]. i3 Statprobe used this paper as a starting point to develop a validation process for define.xml. This paper reviews some of the tools that were utilized, plus takes a look at the needs and requirements of the validation process and checks that can be implemented to ensure a useful and compliant define.xml file for regulatory submission.

INTRODUCTION

Define.xml can be challenging to produce, but once this is accomplished, how do you know it is correct? How is ‘correct’ defined? The contents may be ‘correct’, but the file still may not be ‘compliant’. Much is still unknown or unclear regarding what constitutes a valid define.xml file and conversations indicate that define.xml files received by the FDA are not meeting their expectations. Further guidance will be needed over time to assist companies in creating define.xml files that meet the needs and expectations of regulatory reviewers. Although there is currently not an industry standard set of validation checks for define.xml, some degree of quality control must be undertaken to provide assurance that the file is useful for supporting the datasets within a regulatory submission. Without a functional and informative data definitions file, it may be extremely challenging to navigate through the data and may lead to questions from regulatory reviewers following a submission.

PURPOSE/USAGE OF DEFINE.XML

The FDA’s current Study Data Specifications[3] requests a data definitions file (define doc) which ”describes the format and content of the submitted datasets”. For datasets submitted following CDISC SDTM[4], the agency additionally indicates “The specification for the data definitions for datasets provided using the CDISC SDTM is included in the Case Report Tabulation Data Definition Specification (define.xml) developed by the CDISC define.xml Team”. Although define.xml is not officially required at this time (note: the CBER reviewing divisions are actually beginning to require define.xml for CDISC submissions[5]), it is highly encouraged. Additionally, studies which are to be loaded into the Janus data repository must be accompanied by define.xml[6].

The data definitions file lists each of the submitted datasets as well as the specific contents of each dataset. Within this metadata is information to identify the specific sources of each data value or the methodology for how it was derived, thus documenting the traceability of any particular data value. Bookmarks and hyperlinks allow a reviewer to easily move between components of the document or to the related files (datasets, annotated case report forms, reviewer’s guide, supplemental data definitions file). Use of the define.xml format (following CRT-DDS) allows for additional metadata content to be communicated to the agency and provides a machine-readable data definitions file that can be used with review tools.

The data definitions file, if properly generated and populated, is an extremely valuable tool to aid in reviewing the clinical data. This file essentially provides a roadmap for the submitted data, with full navigational features. It is the GPS device for the reviewers at a regulatory agency. Without it (or with a lesser feature set), someone may decide to take your data for a spin and get lost!

WHY YOU SHOULD VALIDATE DEFINE.XML

Potential issues if define.xml files are not properly validated include:

1) File cannot be viewed properly

   If the file cannot be opened and viewed, then it is essentially useless regardless of how much information may have been included.

2) Contents do not match the datasets they are intended to describe

   If there is a mismatch between the actual dataset contents and those documented within the define doc, then one or more issues may potentially be present:

   - Information is missing for a variable or dataset which is present but not documented
   - Information may be documented for data which does not exist within the submitted datasets
   - The documented metadata is not consistent with the actual dataset contents, e.g. variable type attribute indicates that a variable is numeric, but it is actually character

   Any of these issues may lead to questions, confusion, or concerns regarding the submitted data.

3) Misinformation

   Incorrect or invalid information can lead to concerns regarding the validity of the data.

4) Non-compliant with CRT-DDS

   If the file is not compliant with CRT-DDS, then it potentially cannot be used with standard programs or tools for reviewing the data. Furthermore, it may not provide the features and functions expected for a define.xml file.

The necessity for validation of define.xml is thus related to the purpose and intent of define.xml – if the file cannot be used or does not provide the appropriate information, then it is not fulfilling its purpose and may in fact lead to confusion/misunderstanding or communication of the wrong information. Additionally, the CBER reviewing division notes that "If, the define.xml fails, SDTM cannot be validated until the define file is corrected. The Sponsor/Applicant will be contacted to resolve the errors"[5]. Issues with the define.xml file could thus cause delays for the submission.

VALIDATION CHECKS FOR DEFINE.XML

1) Syntax

   Define.xml should be checked for correct XML syntax. If there are any problems with the syntax, there will likely be problems opening or viewing the file. If you have generated define.xml and it will not open or does not render properly when opened, it is recommended to run the file through a XML syntax checker first.

2) Schema

   The schema defines the structure of the define.xml file and is based on an extension of the CDISC ODM[7]. If the define.xml file does not adhere to the expected schema, it may not work with tools or programs built around this schema. CDISC's white paper entitled "XML Schema Validation for Define.xml"[2] covers this component of validation in great detail.

3) Compliance to CRT-DDS

   Although validation against the schema may also check certain components of adherence to the specification, tools for schema validation do not encompass all rules and requirements of the specification. It is important to check that all required elements and attributes have been included in define.xml and are used according to the specification. Certain elements and attributes additionally have specified controlled terminology that should be used.

4) Contents

The actual metadata contents should be crosschecked to the submitted datasets to ensure consistency, completeness, and accuracy of the metadata. This component of the validation ensures that all data has been appropriately documented and described within the metadata. These checks include crosschecking of dataset and variable attributes between the datasets and define.xml, crosschecking value-level metadata to values present in the datasets and crosschecking code lists to the datasets/source documents as well as checking that all required fields are populated appropriately and using controlled terminology where appropriate.

5) Consistency

Related components within define.xml should be crosschecked to ensure consistency and completeness, such as checking code list type to the actual code list values (i.e. a text type code list should contain text values and variables using the code list should also be text type).

6) Functionality

Bookmarks and hyperlinks should be checked to verify that they are functioning correctly, are appropriately labeled/identified, and point to the correct locations.

TOOLS FOR VALIDATION OF DEFINE.XML

There are many tools available from various providers to help validate define.xml files. Many of these are noted within CDISC's white paper "XML Schema Validation for Define.xml"[2].

i3 Statprobe tested several tools which were available at no additional expense and examined their functionality and ease of use as well as the actual validation aspects. These tools included DefineValidator by Phase Forward (available for download at http://www.phaseforward.com/products/cdisc/), OpenCDISC Validator (available for download at http://www.opencdisc.org/download), and the SAS® Clinical Standards Toolkit (contact your SAS administrator). These tools were ranked for several categories:

Tool	Set-up	Ease of Use	Validation Checks	Validation Reports
DefineValidator	Moderate	Moderate	Not ranked	Moderate
OpenCDISC Validator	Easy	Easy	Excellent	Moderate
SAS Clinical Standards Toolkit	Difficult	Difficult	Excellent	Not ranked

DefineValidator checks define.xml against the schema. It is rated Moderate in Set-up and Ease of Use categories since it requires the use of DOS commands to install and execute the files. The user must open a DOS command prompt window, be comfortable navigating through directories via DOS, and enter the appropriate commands to execute the program and obtain results. An overall Pass/Fail status is summarized within the DOS window once it finishes execution and the user can specify an option to write out a log (which serves as the validation report). The prominent issue with this tool was that Phase Forward does not provide a list of the actual checks performed, thus it is hard to identify how robust the tool actually is. Additionally, one must save the define.xml file being validated within the same folder in which DefineValidator was installed. As it was not practical nor possible to install DefineValidator within every study directory, the tool had to be installed on users' local machines and define.xml files copied over for testing. Additionally, to review any error messages, one must remember to specify an option to write out the 'filtered' define.xml file that was processed by the tool; otherwise, the line numbers referenced in the error messages will not match up to the define.xml file.

OpenCDISC's Validator checks define.xml against the schema and CRT-DDS. It is rated Easy in Set-up and Ease of Use categories as it is very straightforward to download, install, and use. This tool has a nice user interface, allowing the user to click and select the options and file to test. Additionally, OpenCDISC provides a full list of checks which it performs for define.xml. The thoroughness of checks gave this tool an Excellent rating for validation checks. It certainly helped us identify several inconsistencies during our initial testing phases which could then be addressed systematically once we put our define.xml process into production. The reports can be cumbersome to interpret at times, leading to a Moderate rating for the reports category. Several of our define.xml files initially failed in OpenCDISC after passing in DefineValidator, so it seems that OpenCDISC has additional checks beyond that of DefineValidator.

The SAS Clinical Standards Toolkit (CST) was also tested. This tool received a Difficult rating for Set-up as we had to work with our internal procurement and SAS sales contact to procure the software. Although it is free of charge with many SAS packages, we had to complete some paperwork with SAS to obtain and use the CST package. This did not move along as quickly as we would have liked and delayed our ability to test and use it. We then had to work with our IT group to install the toolkit on the server, causing further delays. There are many validation checks built into CST and they are well documented, albeit the documentation can be difficult to find. Unfortunately, this tool is set up to perform most of the validation checks against the internal datasets used to generate define.xml by the tool, so you actually have to generate your define.xml via CST to make use of the checks. Alternatively, you could read your external define.xml file into SAS and transform the data into datasets mirroring the structure and naming used in the generation of define.xml by CST, however due to CST's strict and complex data requirements, it was deemed to be too much effort to try to set up a program to load and transform the define.xml to use with this tool, thus we are unable to comment on the validation reports. CST does have the capability to perform syntax checks on externally generated files, however even to run that set of checks, you still need to set up a specific dataset and create a specific macro call (and it always seems that there are macros that call macros that call macros and they all have to be saved in a specific location). Given that there are several free XML syntax checkers available online (which will not be summarized in this paper) which are very simple to use, we opted not to attempt these checks using CST either.

It is worth noting that DefineValidator was adapted from a feature of the WebSDM^{TM} tool which may be used by the FDA and the CBER reviewing division notes that they will use OpenCDISC Validator to validate define.xml files for CDISC submissions[5].

INCORPORATING VALIDATION INTO THE DEFINE.XML PROCESS

Given all of the types of checks that should be performed for define.xml and the potential problems if there are issues with define.xml, it is critical to incorporate validation into the define.xml process. Validation checks can be incorporated into various steps within the overall process for generating define.xml, from employing checks on your dataset specifications to verifying the XML structure of the generated define.xml. A validation checklist is helpful to ensure that all necessary checks are performed on each define.xml file that is generated. Many of the checks for content and adherence to CRT-DDS can be performed prior to generating define.xml. Standard programs or tools for generating define.xml can incorporate crosschecks between the submission datasets and specified metadata, as well as crosschecks between the various components of the metadata (e.g. Value Level tables back to source variable values). They can also check for the required components and terminology.

If using a standard program or tool to generate define.xml, checks between the datasets and their metadata can be incorporated within that program or tool to look for issues up front, before actually generating define.xml. If issues can be identified up front, this will save time performing QC of the generated define.xml file and reduce the potential for re-runs. Once generated, define.xml could then be read back in (see Lex Jansen's paper "Accessing the metadata from the define.xml using XSLT transformations"[8] if using a SAS-based program or tool) to verify that the metadata was correctly written out in full.

Using standard inputs and a standard program or tool can reduce the time for generating and reviewing define.xml drastically as automated checks can then be built in, reducing the level of manual QC and limiting the potential for human error. A comprehensive design plan and testing can help to assure that the metadata has been thoroughly checked and the resulting file will be complete, accurate, and consistent. It is worth building or acquiring a standard, validated program or tool as it really helps to streamline the process and reduce the time for generating and validating define.xml, which is often done near the end of a project, when timelines are tight. If nothing else, having at least a standard process (even if a manual one) will help to assure that the metadata has been verified and resulting files checked for compliance and accuracy.

Beyond these checks and tools, it is advised to have an independent associate review the file from a submission perspective, to confirm that all appropriate files were generated and packaged, using the appropriate formats and properties, and adhering to submission guidelines. This review also helps to identify any potential inconsistencies in the metadata or any metadata which may be unclear to someone unfamiliar with the datasets.

CONCLUSION

As with any GPS device, proper quality control must be undertaken by the provider to assure users that the maps and information within are correct and up-to-date. With define.xml as the GPS device for navigating through your submission data, it is critical to incorporate validation checks into the process for define.xml to ensure its accuracy, completeness, functionality, and adherence to the CRT-DDS. One should try to think end-to-end and design the process to minimize the work, automate as much as possible, and reduce the potential for errors. This will allow for quicker turnaround, reduced costs, and greater confidence in your define.xml files. Furthermore, that roadmap may just help guide you to a quick approval!

REFERENCES

[1] CDISC. *Case Report Tabulation Data Definition Specification v1.0.*. Retrieved from http://www.cdisc.org/define-xml.

[2] CDISC. *XML Schema Validation for Define.xml*. Retrieved from http://www.cdisc.org/define-xml.

[3] FDA. *Study Data Specifications*. Retrieved from http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM199759.pdf.

[4] CDISC. *Study Data Tabulation Model*. Retrieved from http://www.cdisc.org/sdtm.

[5] FDA. *Submission of Data in CDISC Format to CBER*. Retrieved from http://www.fda.gov/BiologicsBloodVaccines/DevelopmentApprovalProcess/ucm209137.htm.

[6] FDA. *SDTM Validation Specification v.1*. Retrieved from http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM190628.pdf.

[7] CDISC. *Operational Data Model*. Retrieved from http://www.cdisc.org/sdtm.

[8] Jansen, Lex. *Accessing the metadata from the define.xml using XSLT transformations.* Paper CD14, PharmaSug2010. http://www.lexjansen.com/pharmasug/2010/cd/cd14.pdf.

CONTACT INFORMATION

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

Name: Sandra VanPelt Nguyen
Company: i3 Statprobe
E-mail: sandra.vanpeltnguyen@i3statprobe.com