The Smart Dataset-XML Viewer: SDTM Dataset XQuery Validation

Document last update: 2016-04-15

Introduction

As of April 2016, a set of new validation features has been added to the "Smart Dataset-XML Viewer". These features are experimental and will be further improved, enhanced and extended.

Most SDTM/ADaM/SEND validation tools have implementations of the rules that are "black box", i.e. it is completely intransparent how these rules have been implemented. Often these implementations are "overinterpretations" of the standard or published rules (as text). This usually leads to "false positives" where the user does not understand why the value is marked as being a validation violation.

The use of Dataset-XML (and XML in general) has the advantage that the rules can be developed in a language that is at the same time human-readable and machine-executable: XQuery. Overmore, XQuery is a W3C standard, and so completely open and vendor-neutral.

We have developed most of the FDA-SDTM rules in XQuery, and are currently also working on the CDISC-ADaM rules[1]. We will soon also start on the FDA-SEND rules and the rules developed by the PMDA. Also, as soon as the CDISC-SDTM validation rules are released for implementation by CDISC, we will start developing them in XQuery.

You can inspect each of the rules yourself by looking into the directory "Validation_Rules_XQuery" and opening the files that are present there (currently only FDA_SDTM_validation_rules.xml and CDISC_ADaM_validation_rules.xml, which is however not complete yet), the best using an XML editor or a "smart" simple editor such as NotePad++.
Each of the rules is very well documented and you can easily understand how they have been implemented, even when you are not an XQuery specialist.

A snapshot of such an XQuery rule (FDAC107) is given below:

1 We do however need help from ADaM specialists to either help developing these rules in XQuery, testing them on their ADaM files or provide sample ADaM files for testing.

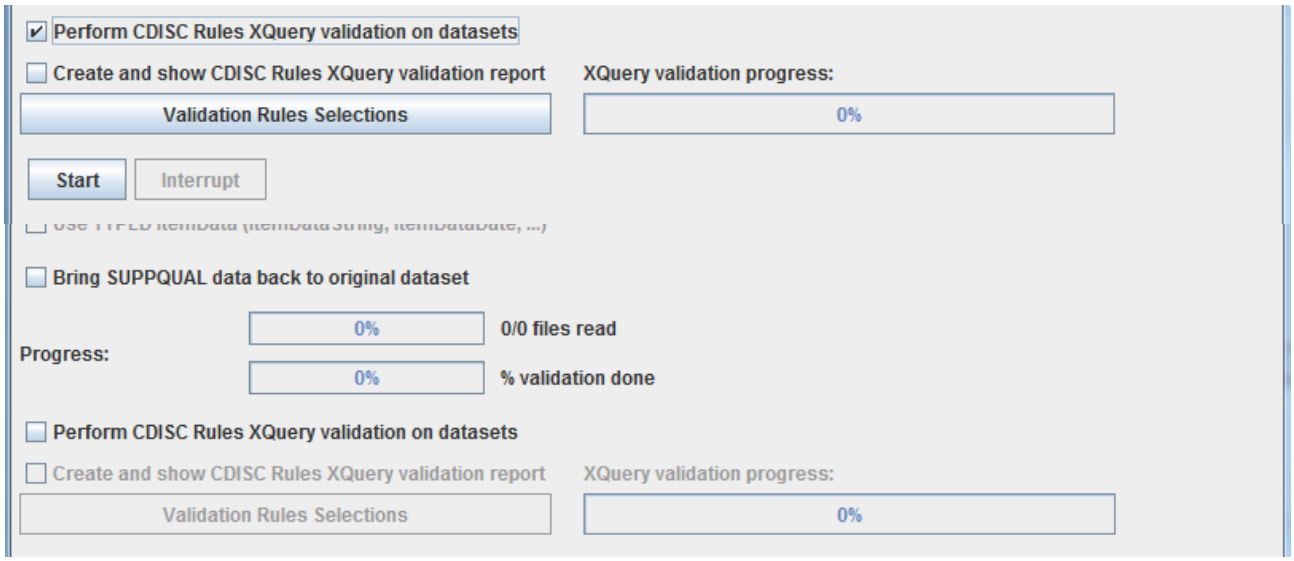
```
(: Rule FDAC107 - --STDTC is after --ENDTC
Start Date/Time of Event, Exposure or Observation (--STDTC) must be less or equal
to End Date/Time of Event, Exposure or Observation (--ENDTC)
:)
xquery version "3.0";
declare namespace def = "http://www.cdisc.org/ns/def/v2.0";
declare namespace odm="http://www.cdisc.org/ns/odm/v1.3";
declare namespace data="http://www.cdisc.org/ns/Dataset-XML/v1.0";
declare namespace xlink="http://www.w3.org/1999/xlink";
(: "declare variable ... external" allows to pass $base and $define from an external programm :)
declare variable $base external;
declare variable $define external;
let $base := '/db/fda_submissions/cdisc01/'
let $define := 'define2-0-0-example-sdtm.xml'

(: iterate over all the datasets :)
for $dataset in doc(concat($base,$define))//odm:ItemGroupDef
    let $name := $dataset/@Name
    let $datasetname := $dataset/def:leaf/@xlink:href
    let $datasetlocation := concat($base,$datasetname)
    (: get the prefix for STDTC and ENDTC from either the domain or the dataset name :)
    let $prefix := if($dataset/@Domain) then $dataset/@Domain
        else substring($name,1,2)
    (: Get the OIDs of the --STDTC and --ENDTC variables :)
    (: We must iterate over all xxSTDTC and xxxSTDTC and xxxxSTDTC :)
    let $stdtcoids := (
        for $a in doc(concat($base,$define))//odm:ItemDef[ends-with(@Name,'STDTC')]/@OID
        where $a = doc(concat($base,$define))//odm:ItemGroupDef[@Name=$name]/odm:ItemRef/@ItemOID
        return $a
    )
    for $stdtcoid in $stdtcoids  (: start iterating over all --STDTCD variables :)
        let $stdtcname :=  doc(concat($base,$define))//odm:ItemDef[@OID=$stdtcoid]/@Name
```

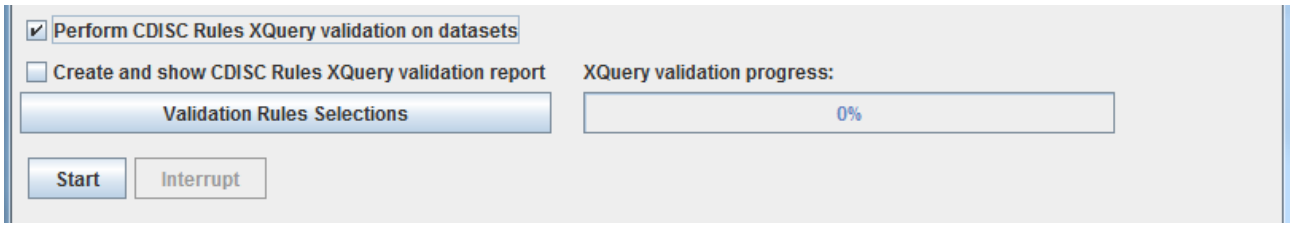
It is important to understand that these rules require a good and correct define.xml file. "Define-XML is leading" as we say, i.e. the define.xml file is "the sponsor's metadata truth". Also, as the FDA does not support define.xml 1.0 anymore anyway, a define.xml 2.0 file is required.

Using the XQuery validation option

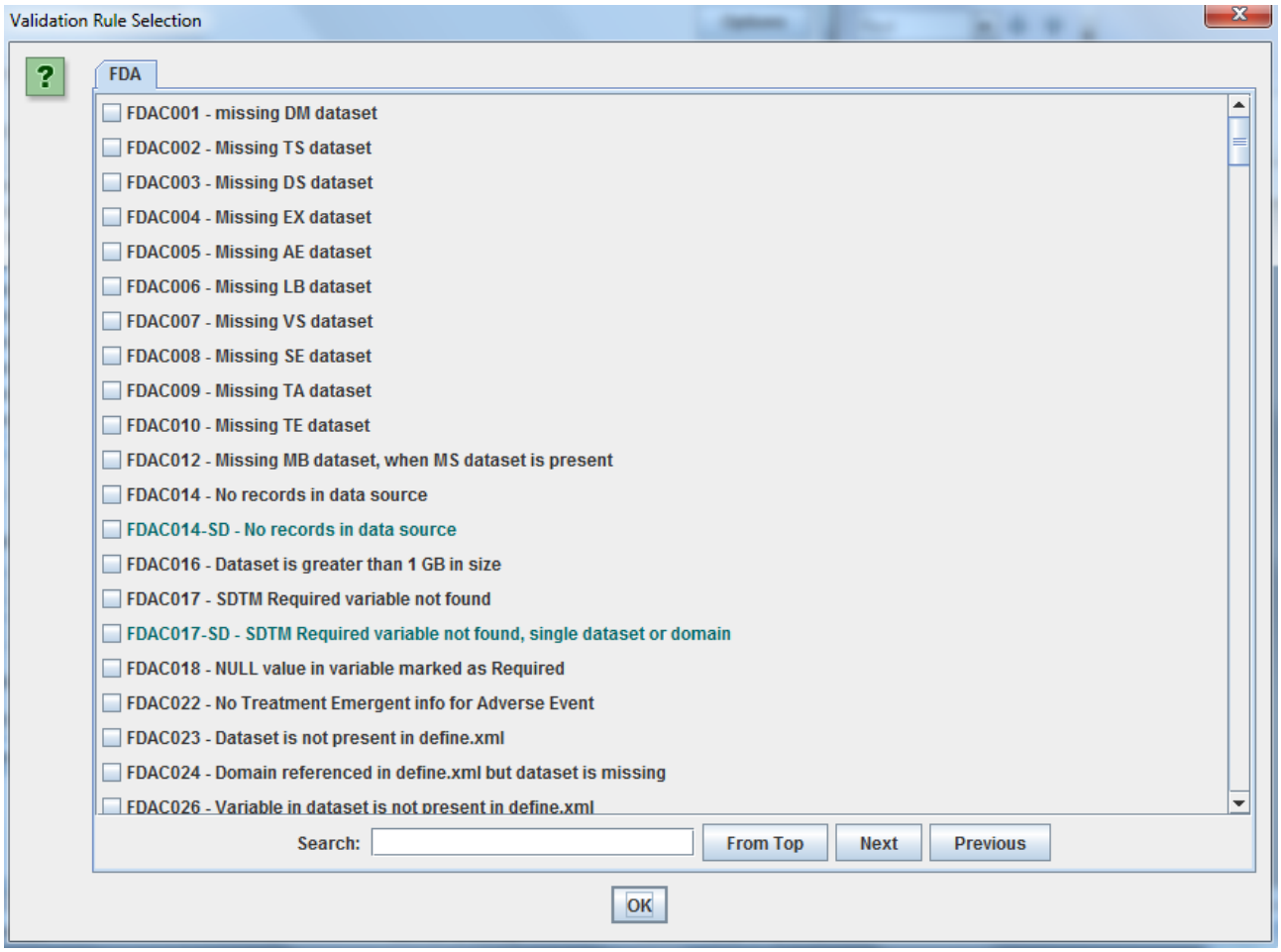
When using the "Smart Dataset-XML Viewer", you could already do simple validation before, using the menu "Options - Validation" and selecting one of the validation options. Additionally, you can now also perform XQuery SDTM validation (currently limited to the FDA-SDTM rules). In order to do so, after loading your define.xml file (version 2.0) and the SDTM datasets, look near the bottom of the screen:



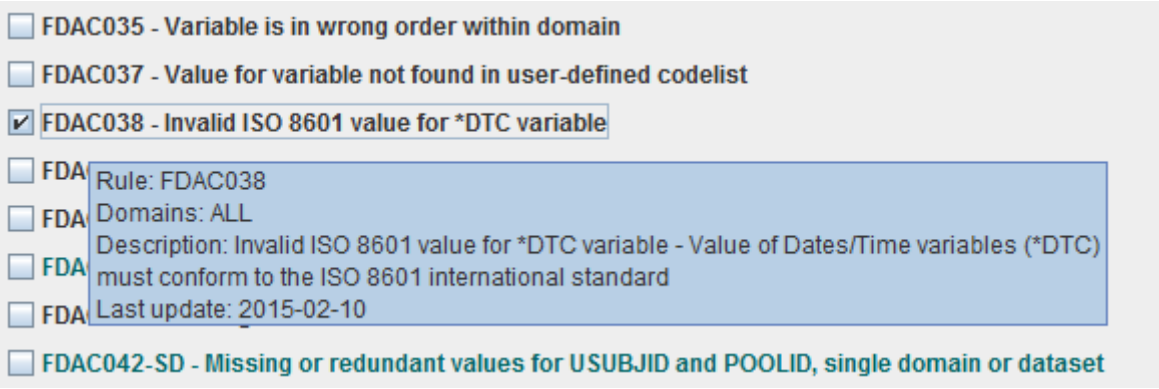
When you then click the checkbox "Perform CDISC Rules XQuery validation on datasets", some of the other widgets become available:



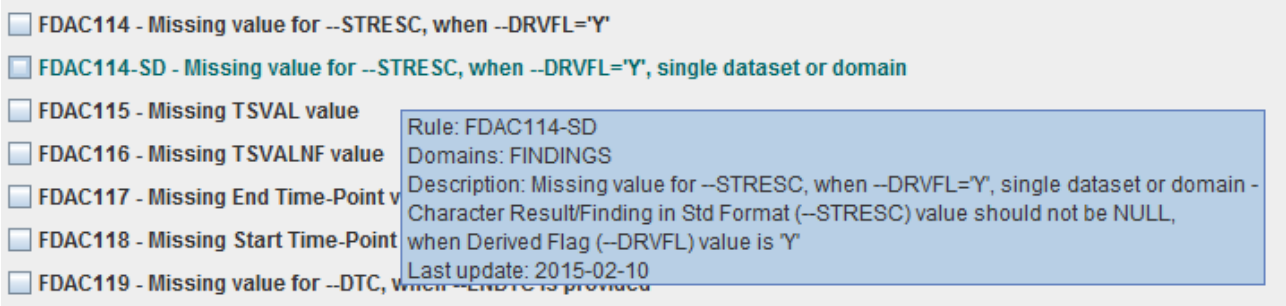
A huge advantage of the current implementation is that the user can decide which rules are being implemented on the files, i.e. you can decide yourself whether your files are tested for some specific rules and not for others. This can be accomplished by using the "Validation Rules Selection" button, leading to a new window with the list of all available rules:



You can now check the rules you want to be implemented during the validation step by checking the corresponding checkbox. Each of the lines has with a tooltip (simply hover over the line with the mouse) giving more details about the rule. For example:



Many of the rules are cross-domain, and so the default is that a rule is applied to all files defined in the define.xml file, even when the file is currently not loaded. This is of course not always efficient. Therefore, you will see that some rules have been implemented twice, once as a generic rule for all domains in the submission, and once to be applied on the loaded files only. These rules have a "-SD" (for single domain) suffix, and have a slightly different color. For example:



One notices that there is a rule "FDAC114", but also a rule "FDAC114-SD".
If the former is selected, the rule FDAC114 will be applied to all the datasets in the directory (even when not loaded). As the execution of this rule is rather computing-intensive, this is not always efficient. In such a case you may decide to apply the rule only to the loaded datasets, by selecting its "domain-specific" rule FDAC114-SD. If selected, the software will iterate over all loaded datasets, check whether the rule is applicable to that dataset (based on information in the define.xml file), and then apply the rule for each applicable dataset that is loaded.

You will find that currently not all FDA-SDTM rules have been implemented. The reason is that some of these rules are simply completely wrong, are not rules but expectations, are not understandable (even by SDTM specialists) or dubious, or have been formulated in such a way that they are not implementable in software. For those that are wrong, not well understandable or dubious, rather than guessing, we choose to simply not implement the rule in such away. The reason is that we want to avoid that false positives are generated, such as is usual when using other validation tools that have implemented the FDA-SDTM rules.

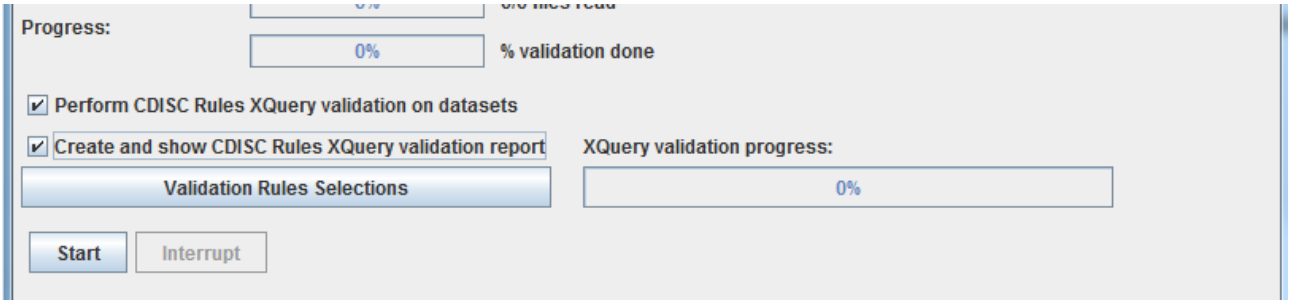
Some of the rules require the use of (RESTful) web services. For example, the system cannot know whether an SDTM variable is "required", "expected" or "permissible". It might even depend on the SDTM version. Rather than writing software for this that reads some files, the better way is to just query a web service that is publicly available[2], and returns the answer within milliseconds. Also other webservices, e.g. delivered by the "National Library of Medicine" (NLM), e.g. looking for RxNorm or SRS information, have been implemented:

2 These webservices are a precursor of the SHARE web services that are currently being developed.

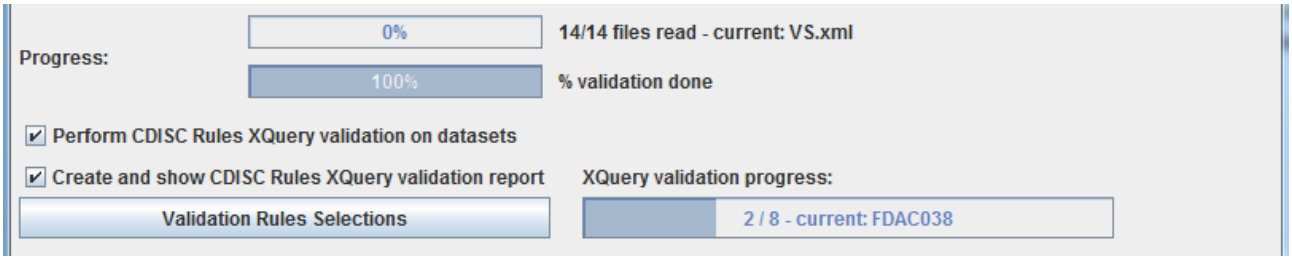
All rules that use a (RESTful) web service are colored blue. They thus require an active internet connection.

After having selected which rules have to be implemented, click "OK" to return to the main window.

In case you want to obtain a report with all the rule violations, also check the checkbox "Create and show CDISC Rules XQuery validation report":



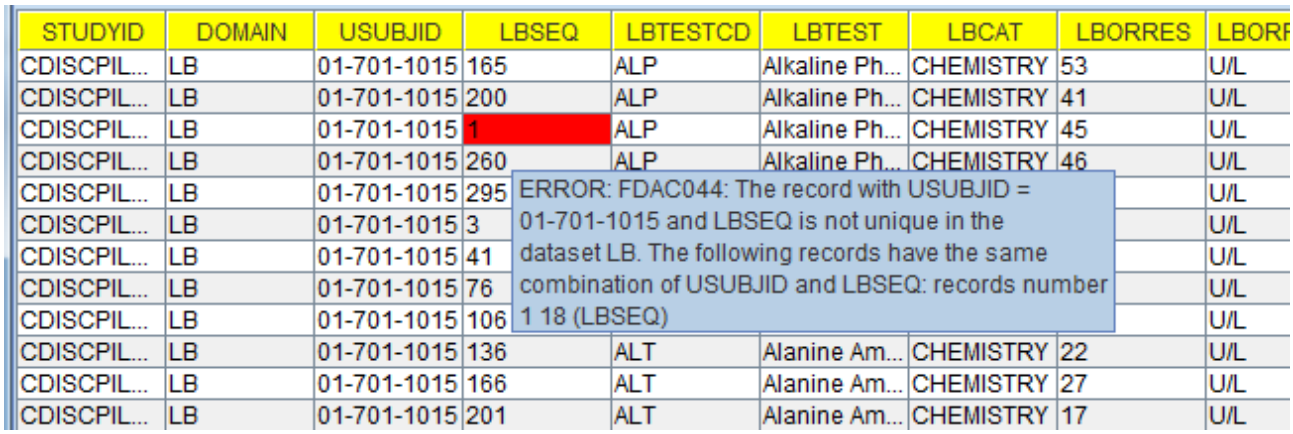
Clicking the "Start" button now starts loading the Dataset-XML files as usual. After this has been accomplished the validation steps, and one can follow the progress on the "XQuery validation progress" bar:



The progress bar shows the progression (number of XQuery validations executed) as well as which rule is currently executed. Please note that rule execution will not be so fast are you might be used with other validation tools, as the XQuery rules first need to be compiled and are executed in a serial way (one rule after the other). Also they currently have not been optimized for speed yet. So it might be useful to first only select a limited set of rules when performing the validations.
This is however counteredacted by the quality of the validation: you should have considerably less "false positives" (if any at all) relative to other available validation tools.

Once finished, one can now inspect the normal SDTM tables.
In case one of the rules has been violated, and there is a record number and variable for the violation, the corresponding cell will either be colored red (in case of an "error") or orange (in case of a "warning"). Also a tooltip on the cell is shown when one hovers the mouse over the cell,

displaying the message. For example:

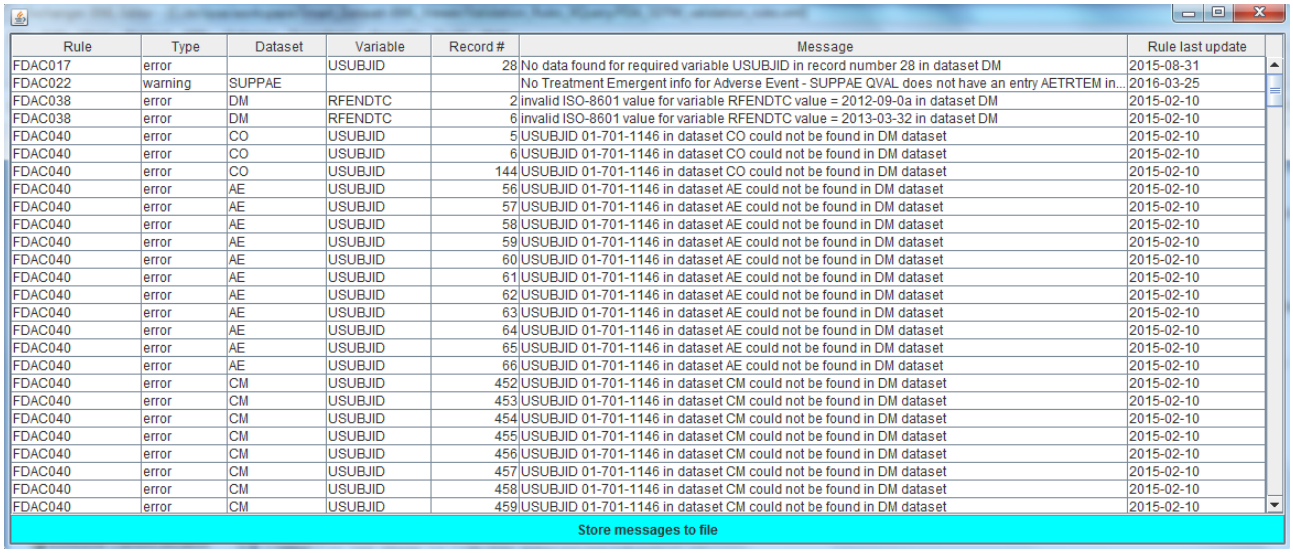


STUDYID	DOMAIN	USUBJID	LBSEQ	LBTESTCD	LBTEST	LBCAT	LBORRES	LBORF
CDISCPIL...	LB	01-701-1015	165	ALP	Alkaline Ph...	CHEMISTRY	53	U/L
CDISCPIL...	LB	01-701-1015	200	ALP	Alkaline Ph...	CHEMISTRY	41	U/L
CDISCPIL...	LB	01-701-1015	1	ALP	Alkaline Ph...	CHEMISTRY	45	U/L
CDISCPIL...	LB	01-701-1015	260	ALP	Alkaline Ph...	CHEMISTRY	46	U/L
CDISCPIL...	LB	01-701-1015	295					U/L
CDISCPIL...	LB	01-701-1015	3					U/L
CDISCPIL...	LB	01-701-1015	41					U/L
CDISCPIL...	LB	01-701-1015	76					U/L
CDISCPIL...	LB	01-701-1015	106					U/L
CDISCPIL...	LB	01-701-1015	136	ALT	Alanine Am...	CHEMISTRY	22	U/L
CDISCPIL...	LB	01-701-1015	166	ALT	Alanine Am...	CHEMISTRY	27	U/L
CDISCPIL...	LB	01-701-1015	201	ALT	Alanine Am...	CHEMISTRY	17	U/L

ERROR: FDAC044: The record with USUBJID = 01-701-1015 and LBSEQ is not unique in the dataset LB. The following records have the same combination of USUBJID and LBSEQ: records number 1 18 (LBSEQ)

In this example, rule FDAC044 "Duplicate value for --SEQ variable" has veen violated. The LBSEQ cell is colored red (as the FDA has defined this as an "error"), and the error message is displayed as a tooltip. Also note that is also indicated that the error can either be in records 1 or 18 (both having the same USUBJID-LBSEQ combination).

If also the checkbox "Create and show CDISC Rules XQuery validation report" has been checked, then also a table with all rule violations will be displayed.



Rule	Type	Dataset	Variable	Record #	Message	Rule last update
FDAC017	error		USUBJID	28	No data found for required variable USUBJID in record number 28 in dataset DM	2015-08-31
FDAC022	warning	SUPPAE			No Treatment Emergent info for Adverse Event - SUPPAE QVAL does not have an entry AETRTEM in...	2016-03-25
FDAC038	error	DM	RFENDTC	2	invalid ISO-8601 value for variable RFENDTC value = 2012-09-0a in dataset DM	2015-02-10
FDAC038	error	DM	RFENDTC	6	invalid ISO-8601 value for variable RFENDTC value = 2013-03-32 in dataset DM	2015-02-10
FDAC040	error	CO	USUBJID	5	USUBJID 01-701-1146 in dataset CO could not be found in DM dataset	2015-02-10
FDAC040	error	CO	USUBJID	6	USUBJID 01-701-1146 in dataset CO could not be found in DM dataset	2015-02-10
FDAC040	error	CO	USUBJID	144	USUBJID 01-701-1146 in dataset CO could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	56	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	57	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	58	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	59	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	60	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	61	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	62	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	63	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	64	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	65	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	AE	USUBJID	66	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	452	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	453	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	454	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	455	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	456	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	457	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	458	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10
FDAC040	error	CM	USUBJID	459	USUBJID 01-701-1146 in dataset CM could not be found in DM dataset	2015-02-10

Store messages to file

As usual with tables in the "Smart Dataset-XML Viewer", it is possible to move columns by dragging and dropping, and to do sorting on the values of any of the columns by simply clicking the column header. For example, when one wants to sort on the record number, one simply clicks on the column header for "Record #" with the following result:

Rule	Type	Dataset	Variable	Record #	Message
FDAC022	warning	SUPPAE			No Treatment Emergent info for Adverse Event - SUPPAE QVAL does not have an entry AETRTEM in..
FDAC055	warning	ARM	TV		Permissible variable ARM has no values for all records in dataset TV.xml
FDAC055	warning	CODTC	CO		Permissible variable CODTC has no values for all records in dataset CO.xml
FDAC038	error	DM	RFENDTC	2	invalid ISO-8601 value for variable RFENDTC value = 2012-09-0a in dataset DM
FDAC040	error	CO	USUBJID	5	USUBJID 01-701-1146 in dataset CO could not be found in DM dataset
FDAC038	error	DM	RFENDTC	6	invalid ISO-8601 value for variable RFENDTC value = 2013-03-32 in dataset DM
FDAC040	error	CO	USUBJID	6	USUBJID 01-701-1146 in dataset CO could not be found in DM dataset
FDAC044	error	AE	AESEQ	15	The record with USUBJID = 01-701-1047 and AESEQ is not unique in the dataset AE. The following...
FDAC044	error	LB	LBSEQ	18	The record with USUBJID = 01-701-1015 and LBSEQ is not unique in the dataset LB. The following...
FDAC017	error		USUBJID	28	No data found for required variable USUBJID in record number 28 in dataset DM
FDAC040	error	EX	USUBJID	30	USUBJID 01-701-1146 in dataset EX could not be found in DM dataset
FDAC040	error	EX	USUBJID	31	USUBJID 01-701-1146 in dataset EX could not be found in DM dataset
FDAC040	error	DS	USUBJID	32	USUBJID 01-701-1146 in dataset DS could not be found in DM dataset
FDAC040	error	DS	USUBJID	33	USUBJID 01-701-1146 in dataset DS could not be found in DM dataset
FDAC040	error	AE	USUBJID	56	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset
FDAC040	error	AE	USUBJID	57	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset
FDAC040	error	AE	USUBJID	58	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset
FDAC040	error	DS	USUBJID	58	USUBJID 01-701-1240 in dataset DS could not be found in DM dataset
FDAC040	error	AE	USUBJID	59	USUBJID 01-701-1146 in dataset AE could not be found in DM dataset

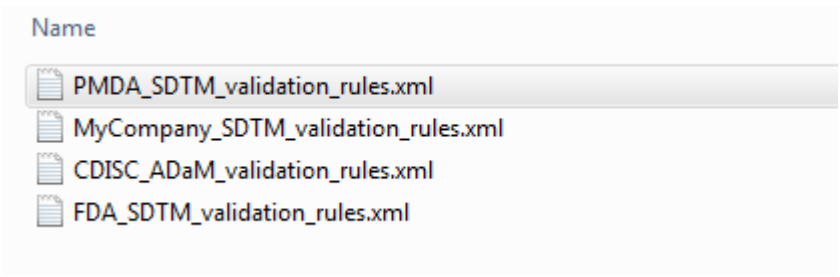
where one also notices that there is a warning that there are no records at all found in the SUPPAE dataset with the "Treatment Emergent" flag (IDVAR=AETRTEM).

One can easily export the content of the table to a file. In order to do so, just click "Store messages to file" and you will be prompted for a file location.
The export format is XML, as this easily allows to reuse the information in other (server) applications (try this with Excel!), or to be stored in databases. Companies can also generate their own stylesheets to show the report again in a browser, but also to generate PDFs, text documents, etc.. The XML used in exactly the same as the one that is generated by the XQuery engine. For example:

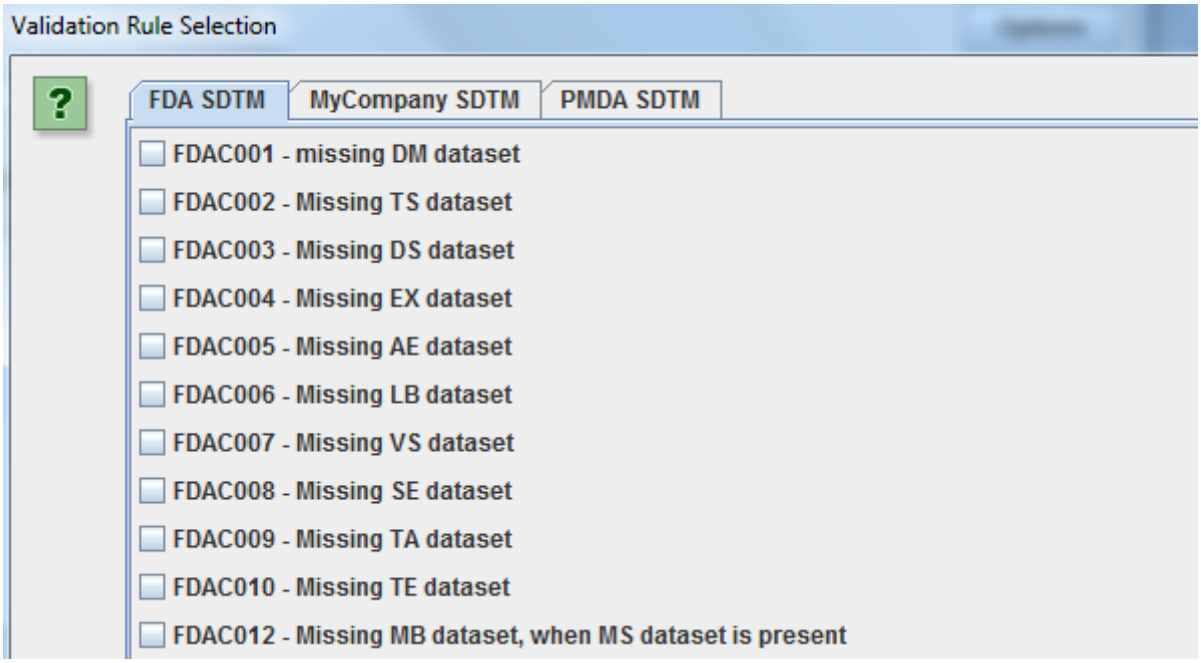
```
<?xml version="1.0" encoding="UTF-8"?>
<XQueryValidationMessages CreationDateTime="2016-04-09T17:42:46.083+02:00">
    <warning rule="FDAC124"
        rulelastupdate="2015-02-10">Collection Study Day CODY variable must be included into dataset, when Collection Study Date/
    <warning rule="FDAC124"
        rulelastupdate="2015-02-10">Collection Study Day CMDY variable must be included into dataset, when Collection Study Date/
    <warning rule="FDAC124"
        rulelastupdate="2015-02-10">Collection Study Day AEDY variable must be included into dataset, when Collection Study Date/
    <warning rule="FDAC124"
        rulelastupdate="2015-02-10">Collection Study Day DSDY variable must be included into dataset, when Collection Study Date/
    <warning rule="FDAC126" dataset="VS" variable="VSDY" rulelastupdate="2015-02-11"
        recordnumber="1451">VSDY is imputed (value=-13) although one of VSDTC (value=2013-05-07) or RFSTDTC (value=) is not a com
    <warning rule="FDAC126" dataset="VS" variable="VSDY" rulelastupdate="2015-02-11"
        recordnumber="1452">VSDY is imputed (value=-13) although one of VSDTC (value=2013-05-07) or RFSTDTC (value=) is not a com
    <warning rule="FDAC126" dataset="VS" variable="VSDY" rulelastupdate="2015-02-11"
        recordnumber="1453">VSDY is imputed (value=-13) although one of VSDTC (value=2013-05-07) or RFSTDTC (value=) is not a com
    <warning rule="FDAC126" dataset="VS" variable="VSDY" rulelastupdate="2015-02-11"
```

Adding your own set of rules

People and companies can easily add their own sets of rules. Just drop the XML file with your own rules as expressed in XQuery in the "Validation_Rules_XQuery" directory. If the filename contains "SDTM", the file will be regarded as a set of SDTM rules, if it contains "ADaM" (case sensitive) it will be regarded as a set of ADaM rules, and similar for SEND. For example, if you have following XML files with XQuery rules in the "Validation_Rules_XQuery" directory:



this will appear in the software as (in case SDTM is selected as the standard):



the tab name being composed from the "originator" attribute in the XML file and the selected standard[3].

You can then select which rules in any of the three tabs will be executed when validating your submission files.

As the "Smart Dataset-XML Viewer" is completely free and open source, no special license or special version of the software is required at all to add company-specific sets of rules.

Please also regularly have a look at the Sourceforge site, as we will regularly update and extend the rule sets, and add new ones (e.g. the PMDA ones are still in development). If a new set is published (e.g. "PMDA ADaM"), just download it and copy it to the "Validation_Rules_XQuery" directory, and it will be available immediately (no more waiting until the next software release).
In order to see how each rule was exactly implemented, you can just inspect the contents of the XML file - no more "black box" implementations.

We are currently also implementing a RESTful web service, allowing the users to check whether new rule sets have been publised or updated when starting up the software, so that the user can decide to update the sets, or decide to continue working with the old set.

Future developments

In the near future, we will further enhance the interactiveness of the software with regard to validation using FDA, PMDA and CDISC development rulesets.
For example when clicking a row in the results table, and using a certain keyboard combination, the applicable row in the "viewer" table will automatically be selected and highlighted, at least when that specific dataset and row was selected.

3 This will probably be changed in future.

It is our aim to keep the "Smart Dataset-XML Viewer" a real "open source" application, without **any** limitations regarding changing the code for own use, redistribution, and use as part of other applications (including commercial ones). As the open source rules describe, users will also always be allowed to branch off their own versions. Together with the superior performance for Dataset-XML CDISC files, this will distinguish the "Smart Dataset-XML Viewer" from other tools for working with SDTM, SEND and ADaM, also with respect to validation.

In order to enable this, we need your support. If you have people in your company who can write Java programs, ask them to contribute to the further development of the software by extending it, and if possible and desired, merge these extensions back into the open source code as published on SourceForge. You (and your company of course) can further contribute by helping developing the FDA, CDISC and PMDA rules in XQuery (which is pretty easy to learn), by testing the rules on your own datasets and reporting your impressions and results back, and by improving or correcting the XQuery implementation of the rules.