Running “Open Rules for CDISC Standards” from within the “Smart Dataset-XML Viewer”

Author: Jozef Aerts
Last update: 2017-04-16

Introduction

The “Open Rules for CDISC Standards” initiative is an initiative from a number of CDISC volunteers (including CDISC staff members) to come to a really open implementation of published rules (either by CDISC itself, the FDA and the PMDA) for SDTM, SEND and ADaM, meaning that the rule implementations are not hidden in “black box” software anymore, but that their implementations are published in both human-readable and machine-executable language.
These human-readable and machine-executable rules can then be used by anyone who desires to do so, either using third party software, or using own developed software. With the rules already available, the development of such validation software is very easy. Several ways of doing so is explained on the website.

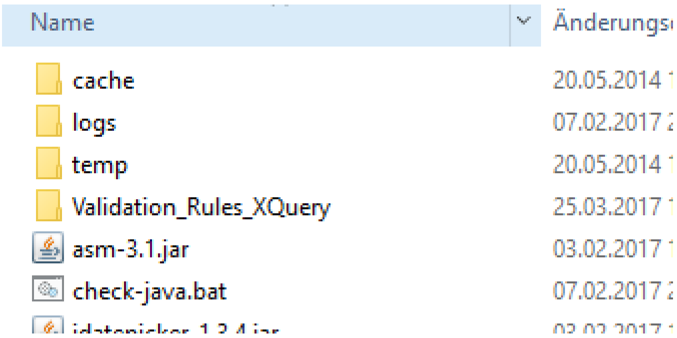
This also enables to easily develop software that does validation without human intervention, for example, software that automatically starts running when new submission files are received by a system. The classic validation software (that is still used by the FDA) does not provide such a functionality anymore.

For those not wanting to write their own software (although it is easy), there are several possibilities. The first one is to store all submissions in a native XML database, and use the query engine of that database. This is explained in another article.
The second method is to use the “Smart Dataset-XML Viewer” which is freely available from the Sourceforge website. As the “Smart Dataset-XML Viewer” is open source, people can also use the source code, extend it, etc.. You can find the source code here.

Remark that the “Open Rules for CDISC Standards” require your (submission) data sets to be in the new CDISC Dataset-XML format, which is already supported by many mapping software providers. If you still use SAS-XPT, you can easily convert your XPT datasets to CDISC Dataset-XML using one of the free conversion tools that are available.

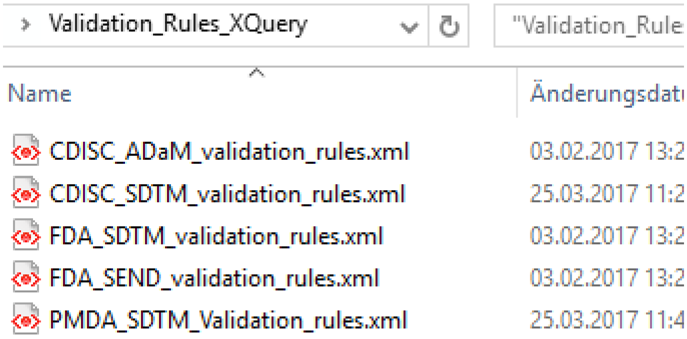
Installation of the rule sets

When you have downloaded and installed the “Smart Dataset-XML Viewer”, have a look at the folder structure:



Name	Änderungs
cache	20.05.2014
logs	07.02.2017
temp	20.05.2014
Validation_Rules_XQuery	25.03.2017
asm-3.1.jar	03.02.2017
check-java.bat	07.02.2017
jdatepicker-1.3.4.jar	03.02.2017

Go into the folder “Validation_Rules_XQuery”. Normally, you will find the following files:



Name	Änderungsdatı
CDISC_ADaM_validation_rules.xml	03.02.2017 13:2
CDISC_SDTM_validation_rules.xml	25.03.2017 11:2
FDA_SDTM_validation_rules.xml	03.02.2017 13:2
FDA_SEND_validation_rules.xml	03.02.2017 13:2
PMDA_SDTM_Validation_rules.xml	25.03.2017 11:4

Each XML file in this folder contains a complete set of rules:

- The ADaM validation checks (v.1.3) as published by CDISC
- The recently published SDTMIG conformance rules (v.1.0) by CDISC
- The SDTM Validator Checks as published by the FDA
- The SEND Validator Checks as published by the FDA
- The SDTM validation rules as published by the PMDA

The most recent versions of these rules sets can always be downloaded from the “Open Rules for CDISC Standards” website. Each rule in the XML file has a date-stamp, so that it is always clear which version of each rule is used[1]. When you download an updated rule set, just copy it in the folder, and it will be ready for use.
Companies can also easily develop their own rule sets in separate files, and just add these files (one for each set of rules) to the folder. Also in such a case, the rules can immediately be used by the “Smart Dataset-XML viewer”.

With the exception of the ADaM rules (about 50% done), we have implemented nearly all the published rules as “open rules”, except for those that:

a) Are not enforceable rules but are expectations
b) Are nonsense (e.g. that each lab result must have a unit)
c) Are confusing or conflict with other rules

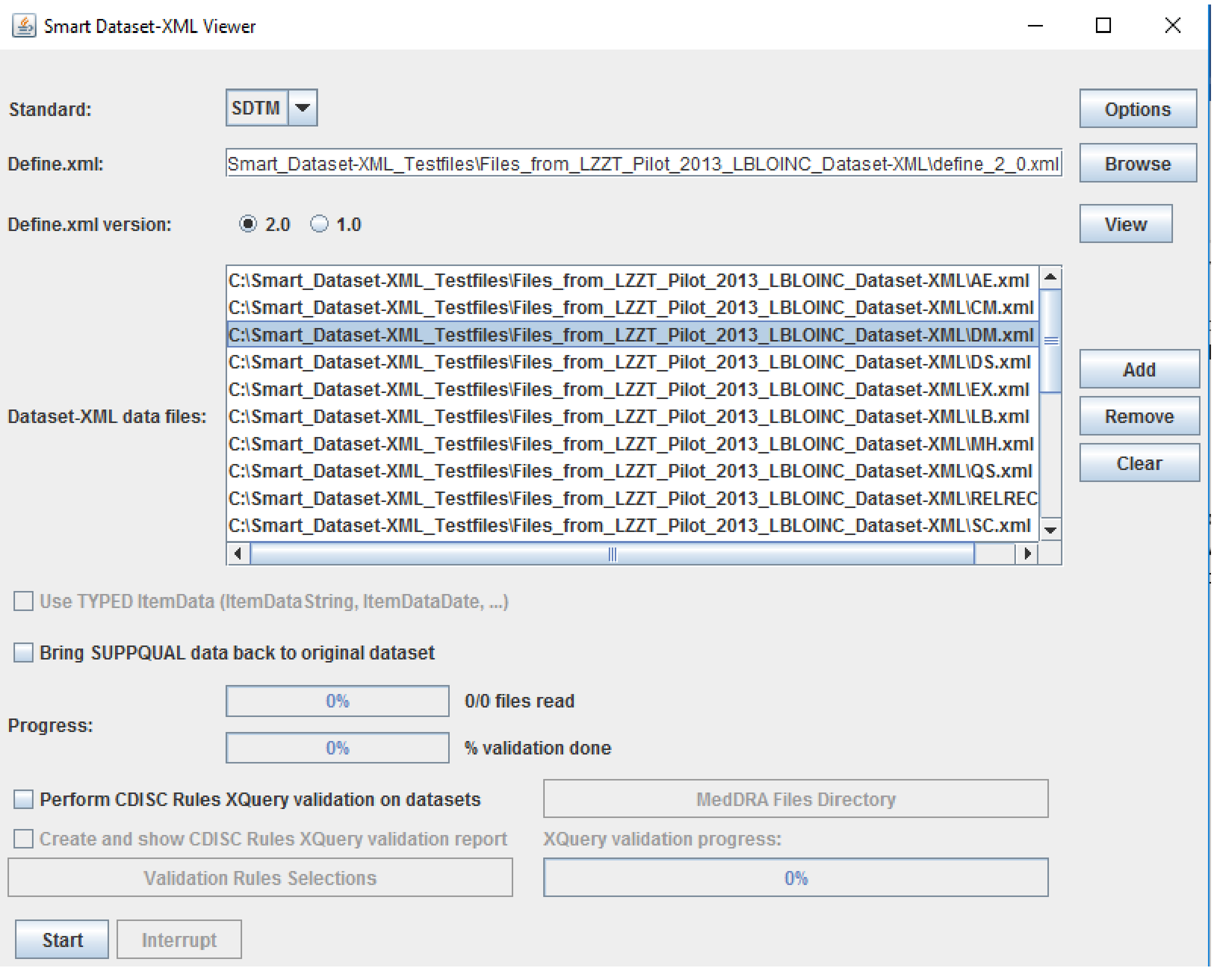
For example, rule FDAC031 “Model permissible variable in added into standard domain” was not implemented as the rule states that it **IS** allowed to add a permissible variable (e.g. “EPOCH”), but at the same time throws a warning. Such kinds of rules of course don’t make sense.

If you are someone with good ADaM knowledge, and would like to help us with the further development of the ADaM validation checks as “Open Rules for CDISC Standards” checks, please let us know, and we will be very happy to have you on board and help you with getting you a “jump start”.

[1] Rule implementations are only updated when it was found they did not function correctly, or when a faster algorithm was found. In such a case, the outcome must be however exactly the same as in the prior version.

Use in the “Smart Dataset-XML Viewer”

After having started the “Smart Dataset-XML Viewer”, first select the location of the define.xml file containing the SDTM, SEND or ADaM metadata[2]. Then select the files that you want to show the data for or want to validate using the “Open Rules for CDISC Standards”. The graphical interface then looks like:



Standard: SDTM

Define.xml: Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\define_2_0.xml

Define.xml version: ● 2.0 ○ 1.0

Dataset-XML data files:
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\AE.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\CM.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\DM.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\DS.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\EX.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\LB.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\MH.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\QS.xml
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\RELREC
C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_LBLOINC_Dataset-XML\SC.xml

□ Use TYPED ItemData (ItemDataString, ItemDataDate, ...)

□ Bring SUPPQUAL data back to original dataset

Progress: 0% 0/0 files read
0% % validation done

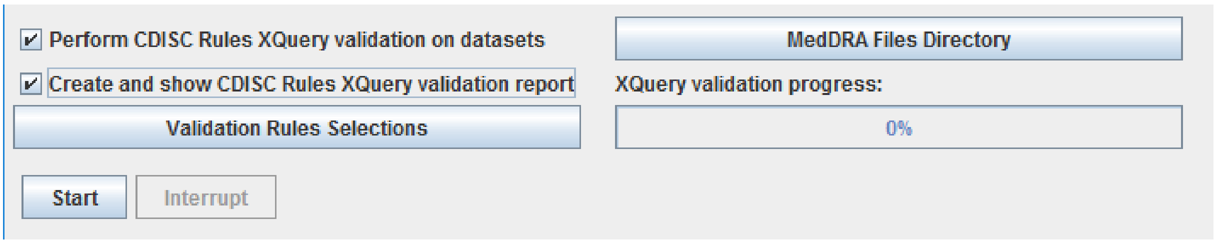
□ Perform CDISC Rules XQuery validation on datasets

□ Create and show CDISC Rules XQuery validation report

XQuery validation progress: 0%

Always ensure you have the DM file in the list, as many of the rules require data from the DM dataset.

Near the bottom, at the left side, you see the checkbox “Perform CDISC Rules XQuery validation on datasets”. Check it if you want to use the validation rules. In case you want to obtain a validation report, also check the “Create and show CDISC Rules XQuery validation report”:



☑ Perform CDISC Rules XQuery validation on datasets

☑ Create and show CDISC Rules XQuery validation report

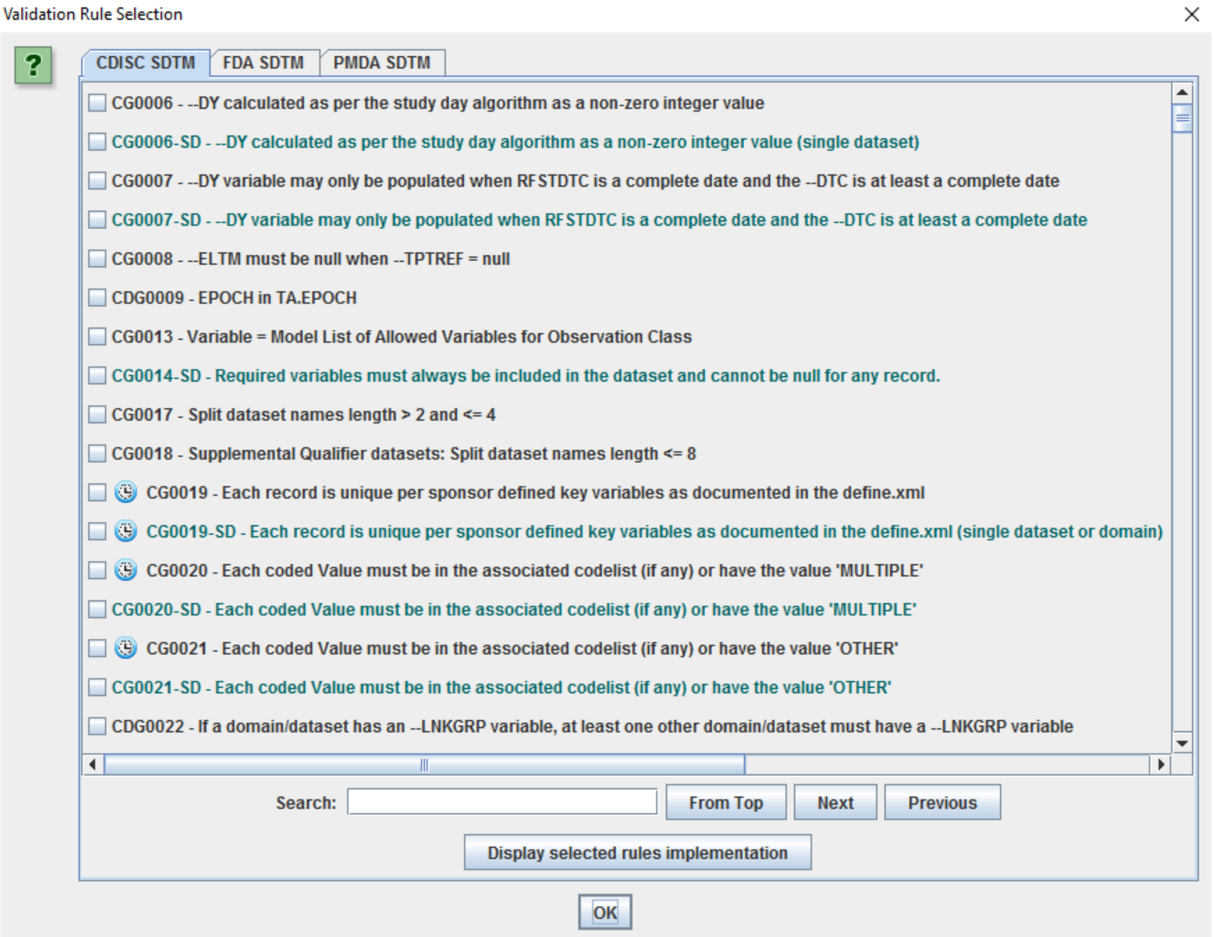
XQuery validation progress: 0%

As you will see, the “Validation Rules Selections” button becomes available, as well as a button “MedDRA files Directory”. The latter allows you to set the directory where you have installed your

[2] The Smart Dataset-XML Viewer both supports define.xml v.1.0 as v.2.0, but the latter is recommended as also the regulatory authorities now require v.2.0.

MedDRA files (“.asc” files). As MedDRA requires a license, we are now allowed to redistribute these files with the software.

The next step is to select the rules you want to be run on the files that you selected before. For example, in case your submission is an SDTM submission, clicking the “Validation Rules Selection” opens a new window with:



Validation Rule Selection

□ CG0006 - --DY calculated as per the study day algorithm as a non-zero integer value

□ CG0006-SD - --DY calculated as per the study day algorithm as a non-zero integer value (single dataset)

□ CG0007 - --DY variable may only be populated when RFSTDTC is a complete date and the --DTC is at least a complete date

□ CG0007-SD - --DY variable may only be populated when RFSTDTC is a complete date and the --DTC is at least a complete date

□ CG0008 - --ELTM must be null when --TPTREF = null

□ CDG0009 - EPOCH in TA.EPOCH

□ CG0013 - Variable = Model List of Allowed Variables for Observation Class

□ CG0014-SD - Required variables must always be included in the dataset and cannot be null for any record.

□ CG0017 - Split dataset names length > 2 and <= 4

□ CG0018 - Supplemental Qualifier datasets: Split dataset names length <= 8

□ 🕓 CG0019 - Each record is unique per sponsor defined key variables as documented in the define.xml

□ 🕓 CG0019-SD - Each record is unique per sponsor defined key variables as documented in the define.xml (single dataset or domain)

□ 🕓 CG0020 - Each coded Value must be in the associated codelist (if any) or have the value 'MULTIPLE'

□ CG0020-SD - Each coded Value must be in the associated codelist (if any) or have the value 'MULTIPLE'

□ 🕓 CG0021 - Each coded Value must be in the associated codelist (if any) or have the value 'OTHER'

□ CG0021-SD - Each coded Value must be in the associated codelist (if any) or have the value 'OTHER'

□ CDG0022 - If a domain/dataset has an --LNKGRP variable, at least one other domain/dataset must have a --LNKGRP variable

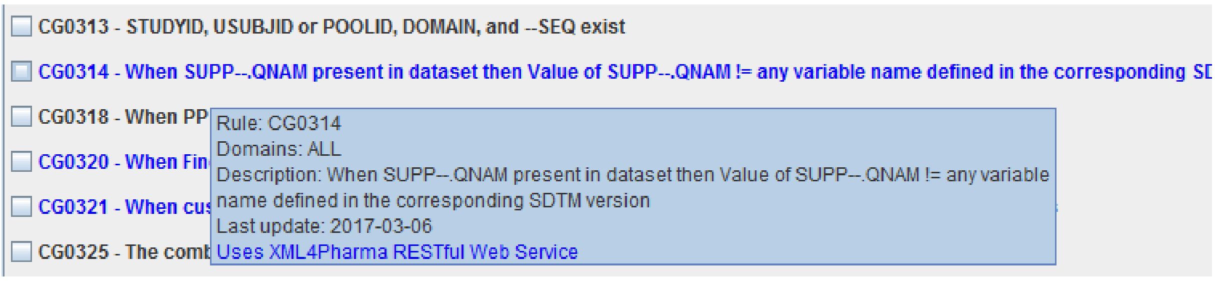
As you have 3 files with rules (3 “rule sets”) in the “Validation_Rules_XQuery” folder starting with “SDTM”, the system displays these 3 sets as 3 tabs in the window. You can then select rules from a single rule set, of make combinations.

You will currently not find a “select all” button as this doesn’t make sense, for several reasons:

a) Some of the rules have 2 implementations, one for “all datasets”, and one for “only the loaded datasets”. The latter have a blue foreground color. It usually are rules that, when applied to all datasets (even the ones that are not loaded) take somewhat (or considerably) more time. Such rules are marked with a “clock” icon, for example rule CG0020.

b) It doesn’t make sense to execute all tests in all cases. For example, once you are sure that none of your “—TEST” variable values has more than 40 characters, it doesn’t make sense to repeat this check over and over again. Instead, you can better concentrate on the rules for which you are expecting problems in your data.

In some cases, you will also find rules that have a dark-blue foreground color.



□ CG0313 - STUDYID, USUBJID or POOLID, DOMAIN, and --SEQ exist

□ CG0314 - When SUPP--.QNAM present in dataset then Value of SUPP--.QNAM != any variable name defined in the corresponding SD

□ CG0318 - When PP

□ CG0320 - When Fin

□ CG0321 - When cus

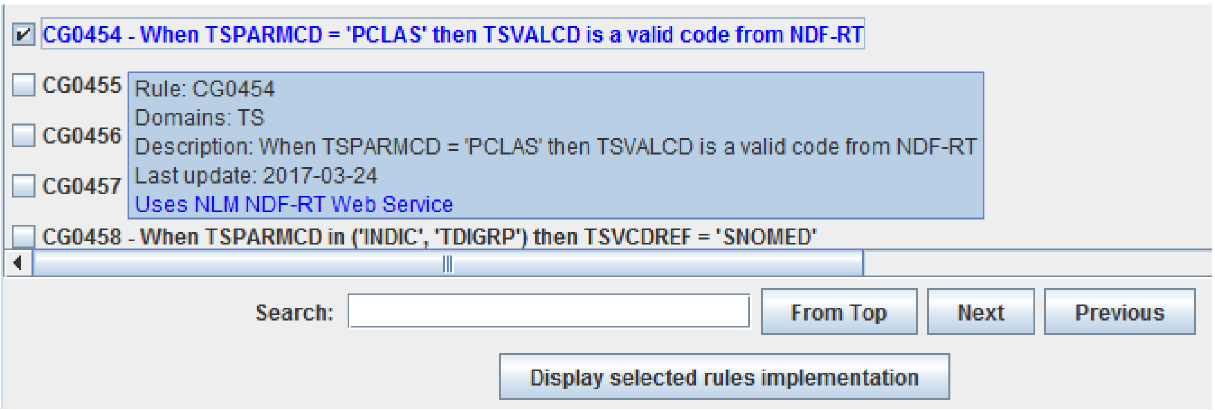
□ CG0325 - The comb

Rule: CG0314
Domains: ALL
Description: When SUPP--.QNAM present in dataset then Value of SUPP--.QNAM != any variable name defined in the corresponding SDTM version
Last update: 2017-03-06
Uses XML4Pharma RESTful Web Service

These are rules that use RESTful web services, either provided by XML4Pharma, by the National Library of Medicine (NLM) or by the FDA itself. Examples are:

- Look up whether a variable is "required", "expected" or "permissible"
- Look up whether a codelist is extensible or not
- Look up whether a given code is really part of a specific codelist[3]
- Look up a LOINC code or an RxNorm or NDF-RT code[4]

These RESTful web services never send any data that contain subject information (like USUBJID, AGE, …). It only sends metadata (like a LOINC code) without any subject identifier, so you can use it without any problems. As our rules are completely transparent, you can even inspect the rule code in order to see what is exactly send to the RESTful web service.
This can be done for any rule using the button "Display selected rules implementation". For example for the rule "When TSPARMCD=PCLASS, then TSVALCD is a valid code from NDF-RT":



☑ CG0454 - When TSPARMCD = 'PCLAS' then TSVALCD is a valid code from NDF-RT

□ CG0455

□ CG0456

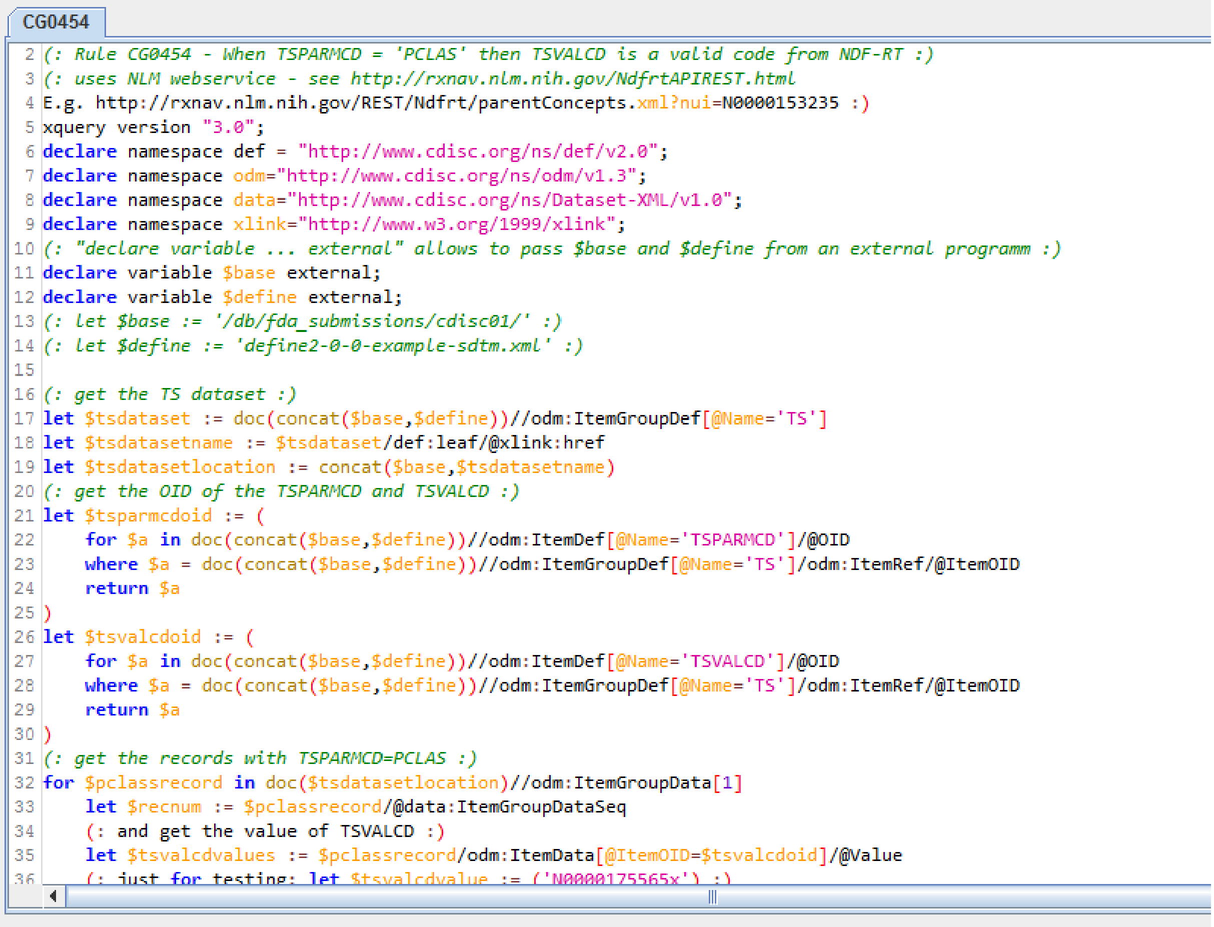
□ CG0457

□ CG0458 - When TSPARMCD in ('INDIC', 'TDIGRP') then TSVCDREF = 'SNOMED'

Rule: CG0454
Domains: TS
Description: When TSPARMCD = 'PCLAS' then TSVALCD is a valid code from NDF-RT
Last update: 2017-03-24
Uses NLM NDF-RT Web Service

Clicking the button "Display selected rules implementation" then displays:

[3] If it is not, and the value is defined as an „extended value" in the define.xml and the codelist is extensible, no (false positive) error is thrown.
[4] Using the NLM web service is much more reliable than a lookup in a local file that is probably outdated. Here is an example.

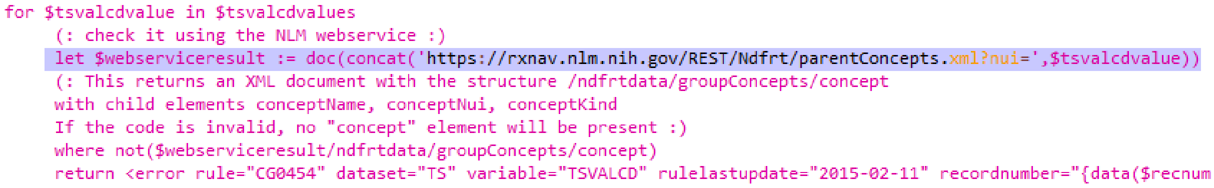


```
(: Rule CG0454 - When TSPARMCD = 'PCLAS' then TSVALCD is a valid code from NDF-RT :)
(: uses NLM webservice - see http://rxnav.nlm.nih.gov/NdfrtAPIREST.html
E.g. http://rxnav.nlm.nih.gov/REST/Ndfrt/parentConcepts.xml?nui=N0000153235 :)
xquery version "3.0";
declare namespace def = "http://www.cdisc.org/ns/def/v2.0";
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' :)

(: get the TS dataset :)
let $tsdataset := doc(concat($base,$define))//odm:ItemGroupDef[@Name='TS']
let $tsdatasetname := $tsdataset/def:leaf/@xlink:href
let $tsdatasetlocation := concat($base,$tsdatasetname)
(: get the OID of the TSPARMCD and TSVALCD :)
let $tsparmcdoid := (
    for $a in doc(concat($base,$define))//odm:ItemDef[@Name='TSPARMCD']/@OID
    where $a = doc(concat($base,$define))//odm:ItemGroupDef[@Name='TS']/odm:ItemRef/@ItemOID
    return $a
)
let $tsvalcdoid := (
    for $a in doc(concat($base,$define))//odm:ItemDef[@Name='TSVALCD']/@OID
    where $a = doc(concat($base,$define))//odm:ItemGroupDef[@Name='TS']/odm:ItemRef/@ItemOID
    return $a
)
(: get the records with TSPARMCD=PCLAS :)
for $pclassrecord in doc($tsdatasetlocation)//odm:ItemGroupData[1]
    let $recnum := $pclassrecord/@data:ItemGroupDataSeq
    (: and get the value of TSVALCD :)
    let $tsvalcdvalues := $pclassrecord/odm:ItemData[@ItemOID=$tsvalcdoid]/@Value
    (: just for testing: let $tsvalcdvalue := ('N0000175565x') :)
```

Don't be frightened by the code! XQuery is very easy to learn. All the XQuery scripts of the "Open Rules for CDISC Standards" are very well documented within the source code. You will probably be able to understand the above code in less than a few minutes.

In our case, the most important lines are:



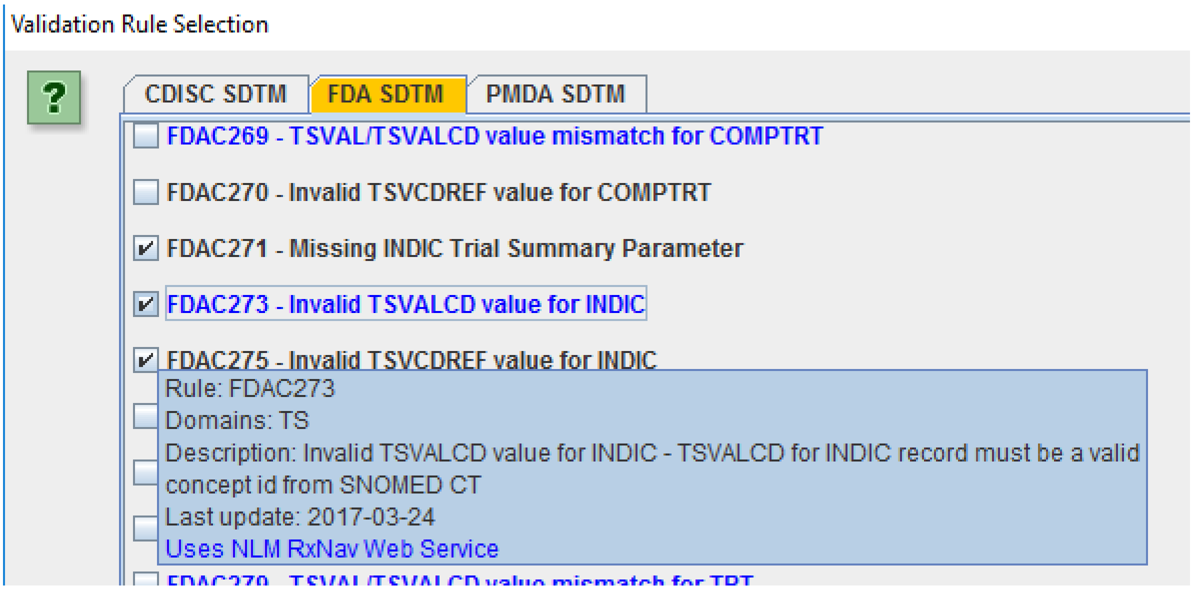
```
for $tsvalcdvalue in $tsvalcdvalues
    (: check it using the NLM webservice :)
    let $webserviceresult := doc(concat('https://rxnav.nlm.nih.gov/REST/Ndfrt/parentConcepts.xml?nui=',$tsvalcdvalue))
    (: This returns an XML document with the structure /ndfrtdata/groupConcepts/concept
    with child elements conceptName, conceptNui, conceptKind
    If the code is invalid, no "concept" element will be present :)
    where not($webserviceresult/ndfrtdata/groupConcepts/concept)
    return <error rule="CG0454" dataset="TS" variable="TSVALCD" rulelastupdate="2015-02-11" recordnumber="{data($recnum
```

In the third line, the base of the NLM web service is concatenated with the value of TSVALCD (when TSPARMCD=PCLAS) and submitted to the NLM server. If the server answer does not contain anything in its XML (path /ndfrtdata/groupConcepts/concept), then the TSVALCD value is not a valid NDF-RT term, and an error is thrown.

After you have selected the rules you want your data be tested for, click "OK" returning you to the main window. You can then still select one or more of many options (that is why the viewer has been given the name "Smart Viewer", using the "Options" button.

To start execution, click the "Start" button. Loading progress is shown, and also the progress of validation of the datasets using the "Open Rules for CDISC Standards". When ready, the datasets are displayed, and in case some of the selected rules were violated, a window will appear with warning and error messages.

For example, when we selected some rules about the trial summary parameter "INDIC":



Validation Rule Selection

□ FDAC269 - TSVAL/TSVALCD value mismatch for COMPTRT

□ FDAC270 - Invalid TSVCDREF value for COMPTRT

☑ FDAC271 - Missing INDIC Trial Summary Parameter

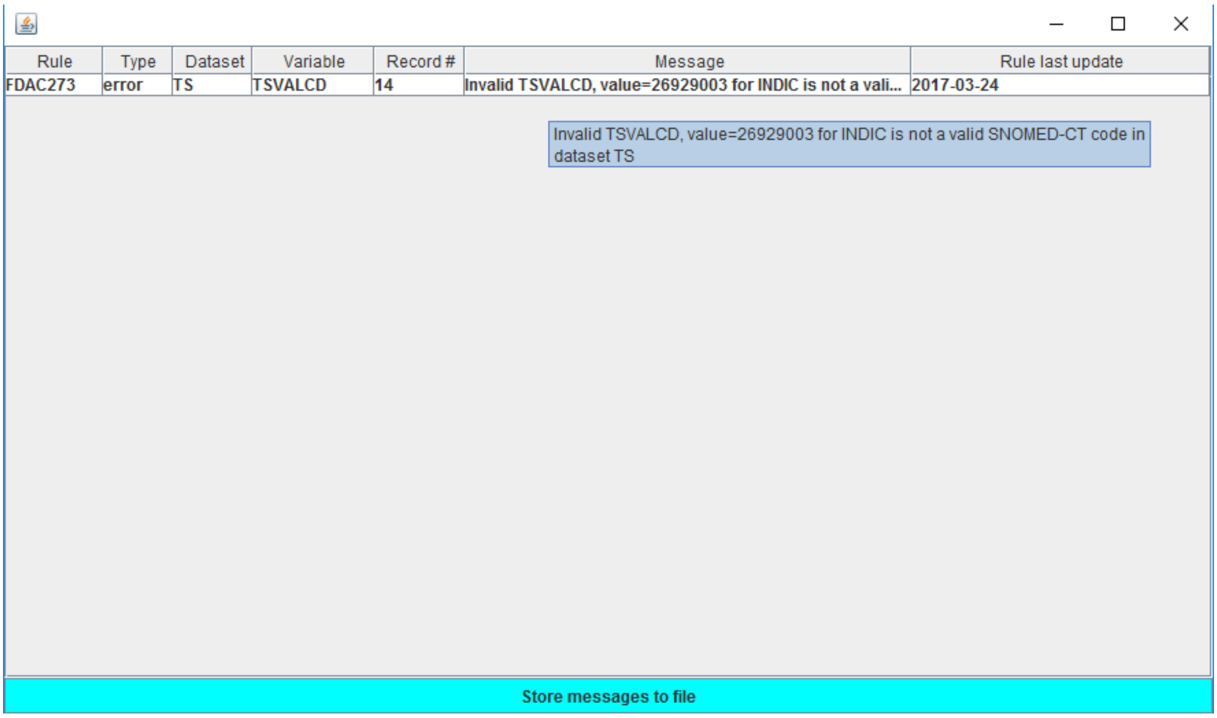
☑ FDAC273 - Invalid TSVALCD value for INDIC

☑ FDAC275 - Invalid TSVCDREF value for INDIC

Rule: FDAC273
Domains: TS
Description: Invalid TSVALCD value for INDIC - TSVALCD for INDIC record must be a valid concept id from SNOMED CT
Last update: 2017-03-24
Uses NLM RxNav Web Service

□ FDAC279 - TSVAL/TSVALCD value mismatch for TPT

And start execution, we might get:



Rule	Type	Dataset	Variable	Record #	Message	Rule last update
FDAC273	error	TS	TSVALCD	14	Invalid TSVALCD, value=26929003 for INDIC is not a vali...	2017-03-24

Invalid TSVALCD, value=26929003 for INDIC is not a valid SNOMED-CT code in dataset TS

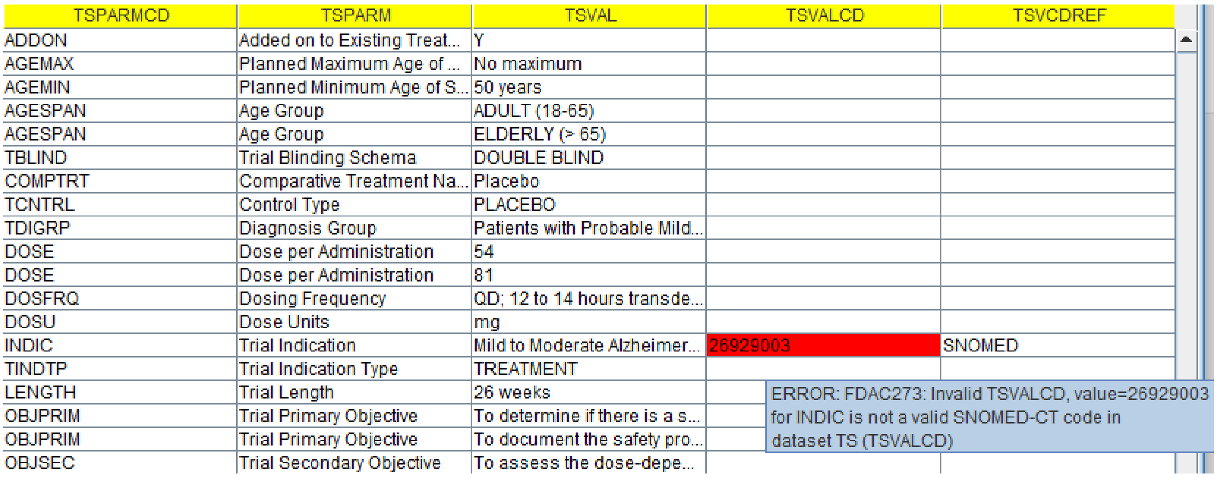
The line in the window states that an error was thrown for rule FDAC273 in record 14 of dataset TS, with the message that an invalid SNOMED code 26929003 was found. This result was obtained by submitting the code 26929003 to the NLM RESTful web service, which answered that it doesn't know this SNOMED code (the correct code is 26929004).

The "store messages" to file then allows to store all validation errors/warnings as XML, which has the great advantage that XML is machine-readable and can be treated by other software systems. This is NOT easily possible with Excel, so an Excel export is not provided. The XML looks like:

```
<XQueryValidationMessages CreationDateTime="2017-04-14T18:09:37.330+02:00">
	<error rule="FDAC273" dataset="TS" variable="TSVALCD" rulelastupdate="2017-03-24"
		recordnumber="14">Invalid TSVALCD, value=26929003 for INDIC is not a valid SNOMED-CT
		code in dataset TS</error>
</XQueryValidationMessages>
```

Also note that as well the table with violations as well as the stored XML contain the information when (date-stamp) the rule implementation was last updated. We also recently implemented a RESTful web service, described at http://xml4pharmaserver.com/WebServices/XQueryRules_webservices.html so that applications can check whether a newer version implementation of any rules is available, and automatically download it and install it in the local software implementation. This has however not been implemented yet in the "Smart Dataset-XML Viewer".

In the viewer, the violation is also depicted in the dataset view itself, i.e. when navigating to the TS dataset in the viewer, one sees:



TSPARMCD	TSPARM	TSVAL	TSVALCD	TSVCDREF
ADDON	Added on to Existing Treat...	Y		
AGEMAX	Planned Maximum Age of ...	No maximum		
AGEMIN	Planned Minimum Age of S...	50 years		
AGESPAN	Age Group	ADULT (18-65)		
AGESPAN	Age Group	ELDERLY (> 65)		
TBLIND	Trial Blinding Schema	DOUBLE BLIND		
COMPTRT	Comparative Treatment Na...	Placebo		
TCNTRL	Control Type	PLACEBO		
TDIGRP	Diagnosis Group	Patients with Probable Mild...		
DOSE	Dose per Administration	54		
DOSE	Dose per Administration	81		
DOSFRQ	Dosing Frequency	QD; 12 to 14 hours transde...		
DOSU	Dose Units	mg		
INDIC	Trial Indication	Mild to Moderate Alzheimer...	26929003	SNOMED
TINDTP	Trial Indication Type	TREATMENT		
LENGTH	Trial Length	26 weeks		
OBJPRIM	Trial Primary Objective	To determine if there is a s...		
OBJPRIM	Trial Primary Objective	To document the safety pro...		
OBJSEC	Trial Secondary Objective	To assess the dose-depe...		

ERROR: FDAC273: Invalid TSVALCD, value=26929003 for INDIC is not a valid SNOMED-CT code in dataset TS (TSVALCD)

Conclusions

This document shows how the "Open Rules for CDISC Standards" can be installed and used in the "Smart Dataset XML Viewer". The advantages of using the "Open Rules for CDISC Standards" are obvious: they are completely transparent, updates do not at all require an update of the software (no more "waiting for the next release"[5]), they use RESTful web services instead of locally installed files (except for MedDRA, who still refuses to provide/allow such services), and the risk of "false positives" is minimal. Although their use is demonstrated here in the "Smart Dataset-XML Viewer", anyone can write his own validation software and still use the "Open Rules for CDISC Standards"

Interested in contributing in one way or another? Just contact us!

[5] When bugs are reported, they usually are fixed and an update made available within a few hours.