Open Rules for CDISC Standards

FDA SDTM Validation Rules 2019

Release Notes 2019-09-15

Introduction

This document contains the release notes for the "Open Rules for CDISC Standards" implementation of the FDA SDTM validation rules 2019.

Open Rules for CDISC Standards aims to provide validation rules for CDISC standards in a format that is completely open, and is as well human-readable as well as machine readable. This means that every user or implementer can check how the rule has been implemented. This in contradiction with rule implementation from other providers, which are mostly "black box".

Open Rules for CDISC Standards is currently being implemented in the XQuery language, a language that is easy to write, easy to read and understand, and can be implemented on as well XML as JSON files. As the FDA still mandates the use of the completely outdated SAS Transport 5 format ("XPT files"), submission files need to be transformed first into the CDISC Dataset-XML format, which is a CDISC standard.
There are many tools and scripts available for doing so, which are documented on the CDISC website. Also the "Smart Submission Dataset Viewer", a free and completely open source software, can easily be used to transform XPT files into the modern CDISC Dataset-XML format.

Open Rules for CDISC Standards is NOT software: it is a set of human- and machine-readable scripts in the XQuery language which can be used by ANY modern software, using any modern software language (Java, C#, Python, SAS, …) on any possible platform (Windows, Linux, Unix, McIntosh, …). Implementors can develop their own software for working with Open Rules for CDISC Standards. One can however also use the "Smart Submission Dataset Viewer" to validate CDISC submission files. Open Rules for CDISC Standards is also perfectly suitable for batch and cron implementation. For example, implementors can run all the rules over night on their newly created or updated submission files.

As **Open Rules for CDISC Standards** is independent of software, updates of rules do not require a software update. It suffices to replace the file(s) with the XQuery scripts. As such, each rule script has a "last-update" attribute allowing versioning of the rules. This also means that in case a bug in a script has been reported, an update of the rule can be developed and published within hours or at maximum within a few days. There is no need to wait for 2-3 years for a new software release as is the case with other validators.

Role of define.xml

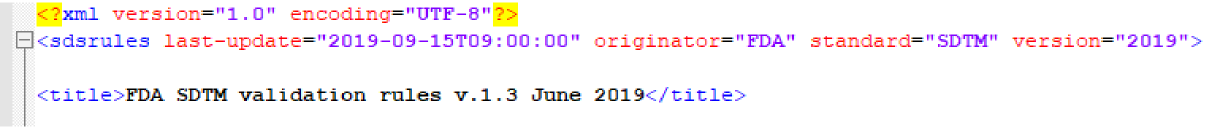
The "define.xml" file, as standardized by CDISC, contains the metadata of the submission. As such, it is the "sponsor's truth" of the submission. This means that when doing validation, a define.xml must be provided, and is expected to be of good quality. Therefore, it is not a bad idea to first do a define.xml validation. Several vendors provide define.xml validation software or even scripts. An example is the "Define.xml Checker" from XML4Pharma.
Almost every Open Rules for CDISC Standards script first reads the define.xml file.

The FDA SDTM 2019 validation rules

The current release concentrates on the FDA SDTM validation rules 2019.
Most of the rules from the by the FDA published Excel file have been implemented in our XQuery scripts. As will be explained, some of the rules have NOT been implemented, as they are nonsense (e.g. rule SD0029, stating that –STRESU must be populated when -STRESC is populated), are expectations rather than a rule, or simply cannot be technically implemented.
These non-implemented rules are documented further on in these release notes.

Metadata for the validation rules

For each set of validation rules, organized by originator (FDA, PMDA, CDISC), standard (SDTM, SEND, ADaM) and version, the rules come as an XML file. In the header of the XML file, the date/time of the last update of the file, the originator, standard to which the rules apply and the rules version (from the originator) is provided:



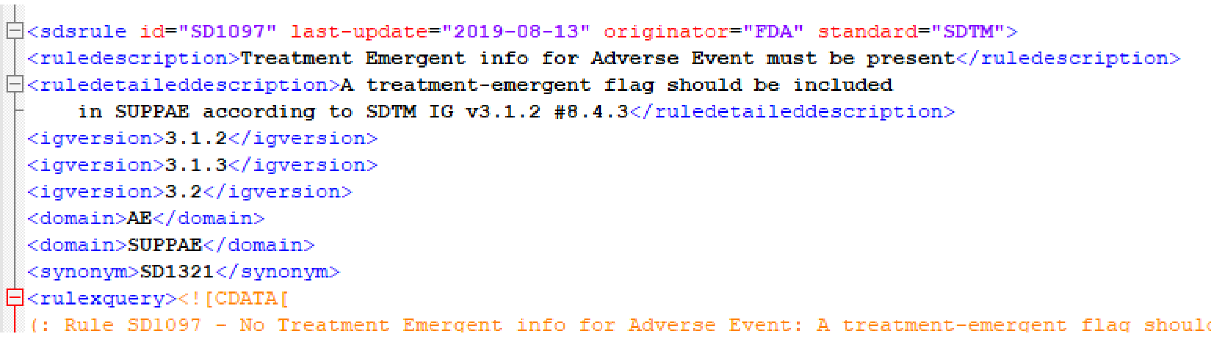
```
<?xml version="1.0" encoding="UTF-8"?>
<sdsrules last-update="2019-09-15T09:00:00" originator="FDA" standard="SDTM" version="2019">

<title>FDA SDTM validation rules v.1.3 June 2019</title>
```

Also a title is provided.

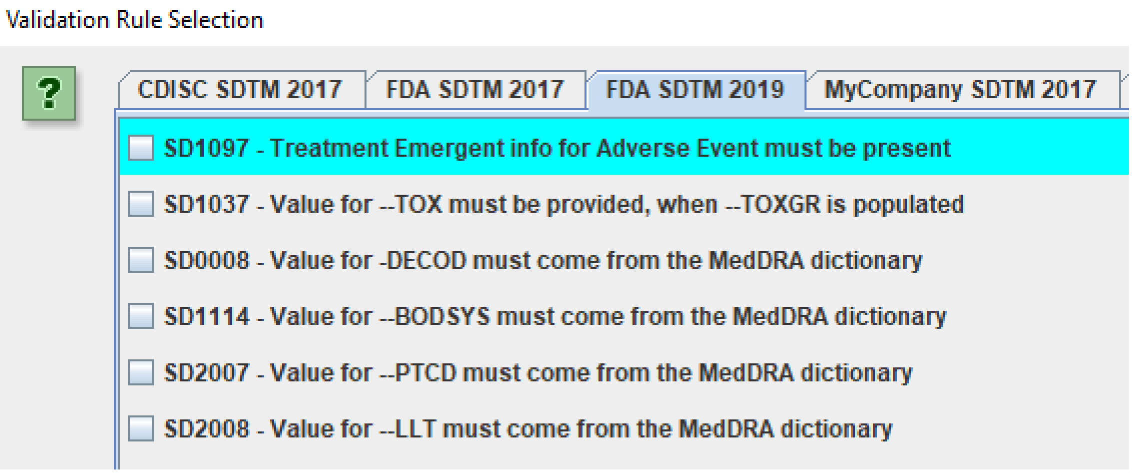
These metadata can be used to check for updates, to provide a title in software, and for many other purposes.

For each separate rule, also a set of metadata is provided. For example, for FDA rule SD1097:



```
<sdsrule id="SD1097" last-update="2019-08-13" originator="FDA" standard="SDTM">
<ruledescription>Treatment Emergent info for Adverse Event must be present</ruledescription>
<ruledetaileddescription>A treatment-emergent flag should be included
    in SUPPAE according to SDTM IG v3.1.2 #8.4.3</ruledetaileddescription>
<igversion>3.1.2</igversion>
<igversion>3.1.3</igversion>
<igversion>3.2</igversion>
<domain>AE</domain>
<domain>SUPPAE</domain>
<synonym>SD1321</synonym>
<rulexquery><![CDATA[
(: Rule SD1097 - No Treatment Emergent info for Adverse Event: A treatment-emergent flag shoul
```

The ID of the rule is provided, the date of the last update of the script implementation, the originator of the rule, and the standard to which the rule applies.
Then a short rule description (element "ruledescription" is provided). Remark that the text does NOT come from the Excel file published by the FDA. We decided to provide such a short description of the rule for the use in software where implementors can select which rules they want to have checked for. For example, in the "Smart Submission Dataset Viewer":


Validation Rule Selection

□ SD1097 - Treatment Emergent info for Adverse Event must be present

□ SD1037 - Value for --TOX must be provided, when --TOXGR is populated

□ SD0008 - Value for -DECOD must come from the MedDRA dictionary

□ SD1114 - Value for --BODSYS must come from the MedDRA dictionary

□ SD2007 - Value for --PTCD must come from the MedDRA dictionary

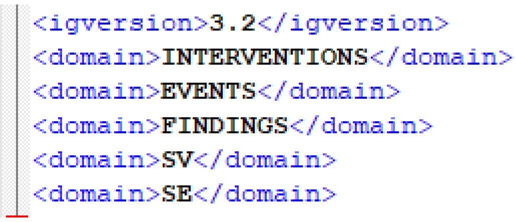
□ SD2008 - Value for --LLT must come from the MedDRA dictionary

The rule as published by the FDA in their Excel file is found in the element "ruledetaileddescription". The latter is essentially normative, this although there are many problems with them:

- Improper use of the wording "should" (which is an expectation), mixed/random use of the wording "must" and "should"
- Lack of separation of the precondition ("when …") and the condition itself ("… then …")
- Bad English
- Unclear description, open for different interpretations ("wiggle room").

In the "igversion" elements, the versions of the SDTM-IG are provided. This allows to select rules for execution depending on the version of the SDTM-IG used for creating the SDTM submission files. The information comes from the Excel file provided by the FDA.

The "domain" elements contain the SDTM domains to which the rule applies. In case the rule applies to a full class of domains, the class name is given. For example:



```
<igversion>3.2</igversion>
<domain>INTERVENTIONS</domain>
<domain>EVENTS</domain>
<domain>FINDINGS</domain>
<domain>SV</domain>
<domain>SE</domain>
```

If the rule applies to all domains, "ALL" is provided.
Also this information comes from the Excel file provided by the FDA.

In some cases, one will also find an element "synonym". This means that the rule is an exact copy of another rule, e.g. of an older version of the FDA rules, or of a CDISC rule. We have however not searched for such "synonyms", i.e. this information may be incomplete.

The rules scripts

The next part is the script in XQuery itself (element "rulexquery"). Implementations that use Open Rules for CDISC Standards scripts can easily extract each of the scripts from the XML.
In the most cases, the script first reads the define.xml file. Its location is passed as a set of two parameters, "$base" which is the directory in which the submission files can be found, and "$define" providing the name of the define.xml file (usually "define.xml"). Remark that the "base" is essentially a URL, so if the define.xml is stored in a file system, one needs to put file:/// in front. For example, for the submission files located in the directory/folder "C:\Smart_Dataset-XML_Testfiles\Files_from_LZZT_Pilot_2013_Dataset-XML_OK\",

one needs to provide:
file:///C:/Smart_Dataset-XML_Testfiles/Files_from_LZZT_Pilot_2013_Dataset-XML_OK/, also replacing the backward slashes by forward slashes.
The reason for this is that the define.xml could also be an internet or intranet file (e.g.: http://mycompany/mysubmissions/submission_007/define.xml" or a define.xml in a native XML database. It could also be a query to a RESTful web service.

Although the FDA does not provide "severities" for their rules (according to our communication with the FDA due to "the severity codes were removed as these concepts do not fit with our current FDA Validator process"), we decided to add a severity level, which can be "info", "warning" or "error".
The reasons we did severity levels are:

- Sponsors expect severity levels to be able to judge whether a violation must be corrected or can be documented in the "Study Data Reviewers Guide"
- Some of the rules are expectations rather than a real rule. In such cases, we assigned the severity "warning" or even "info"

For the moment, we mostly followed the assignment of Pinnacle21[1], in some cases "grading down" the severity where we felt the P21 assignment is wrong.
We will however completely review the "severity" assignments, if possible in cooperation with the FDA, and if necessary, publish an update.

For the error/warning/info message, we did NOT use the messages from the Excel file from the FDA, as we found that these messages are confusing, badly written (e.g. the wording "should", should never be used), have insufficient detail, and in some cases, even contradict with the description of the rule itself. We are sure that our messages are much more precise and provide better detail than the messages provided in the FDA Excel file.

In the metadata for the error/warning/info message, also the date of the last update of the script is provided, allowing version management.

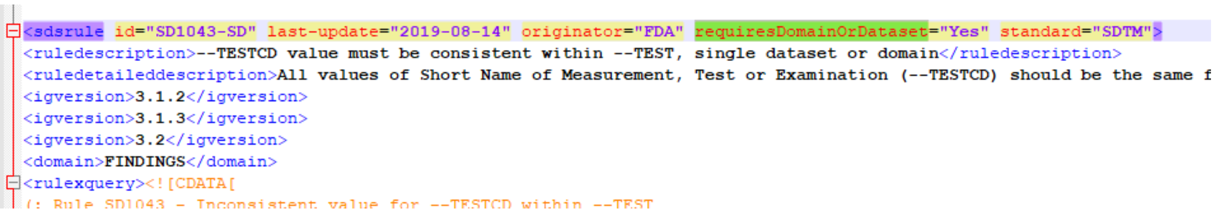
Two "flavors" of the rules

One will notice that some rules have two "flavors":

- "All in" implementation: the rule will iterate over all applicable datasets. For example, if the rule applies to the "FINDINGS" class, it will iterate over all the datasets for which the define.xml has the attribute def:Class="FINDINGS".
  In your own implementation, this may not always be what you want. You might have some datasets that are not complete yet in one way or another, and you only want to apply the rule to a limited set of datasets that YOU decide on.
  In such a case, you can use the "Single Datasets" implementation
- In the "Single Dataset" implementation, you need to base the name of the dataset. For example, if you only want to apply the rule to the datasets LBCHEM and LBURIN, from within your own software implementation, you call the scripts twice, once with the parameter $datasetname being "LBCHEM" and once with the parameter $datasetname being "LBURIN". The script will then execute twice, once on the LBCHEM dataset and once on the LBURIN dataset.

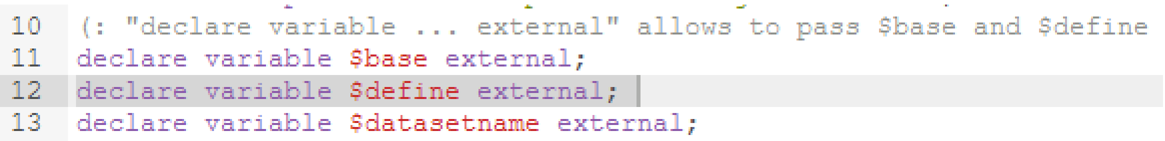
[1] Remark that the "FDA severity" in P21 is NOT an assignment by the FDA (as was communicated by the FDA to us), but an own assignment of P21.

The rules that are "all in" have a normal FDA-ID, e.g. "SD1043", the rules that are "Single Dataset" implementation, have the FDA-ID plus the suffix "-SD". They can also easily be recognized by the additional attribute 'requiresDomainOrDataset="Yes"'. For example, for rule "SD1043-SD":



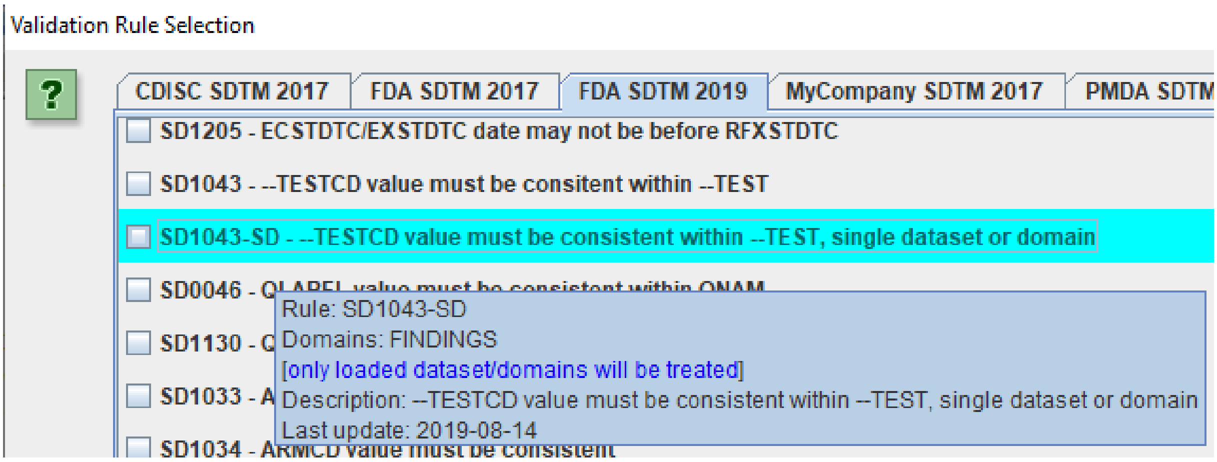
```
<sdsrule id="SD1043-SD" last-update="2019-08-14" originator="FDA" requiresDomainOrDataset="Yes" standard="SDTM">
<ruledescription>--TESTCD value must be consistent within --TEST, single dataset or domain</ruledescription>
<ruledetaileddescription>All values of Short Name of Measurement, Test or Examination (--TESTCD) should be the same f
<igversion>3.1.2</igversion>
<igversion>3.1.3</igversion>
<igversion>3.2</igversion>
<domain>FINDINGS</domain>
<rulexquery><![CDATA[
(: Rule SD1043 - Inconsistent value for --TESTCD within --TEST
```

In the script itself, one will then find the definition of the variable $datasetname as being "external", i.e. the script expects that the name of the dataset will be passed from the calling application.:



```
(: "declare variable ... external" allows to pass $base and $define
declare variable $base external;
declare variable $define external;
declare variable $datasetname external;
```

In implementations such as the "Smart Submission Dataset Viewer" one will often be able to choose between the two "flavors":



□ SD1205 - ECSTDTC/EXSTDTC date may not be before RFXSTDTC

□ SD1043 - --TESTCD value must be consitent within --TEST

□ SD1043-SD - --TESTCD value must be consistent within --TEST, single dataset or domain

□ SD0046 - QLABEL value must be consistent within QNAM

□ SD1130 - Q

□ SD1033 - A

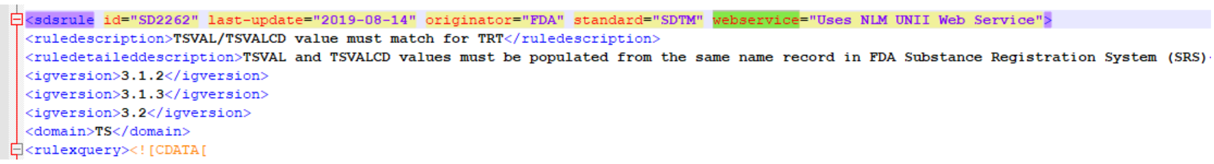
□ SD1034 - ARMCD value must be consistent

Rule: SD1043-SD
Domains: FINDINGS
[only loaded dataset/domains will be treated]
Description: --TESTCD value must be consistent within --TEST, single dataset or domain
Last update: 2019-08-14

Use of RESTful web services

Using files for checking SNOMED-CT, UNII, LOINC codes, … is completely outdated. There is huge problem in other validation tools with providing the most recent version of such coding systems. The modern method is to use a RESTful web service, e.g. from the National Library of Medicine (NLM). Such RESTful web services almost always provide the latest (or versioned) version of the coding system, and no unpractical maintenance of local files is necessary.
The Open Rules for CDISC Standards use RESTful web services as much as possible. It uses RESTful web services from the NLM (a trustful source), from RESTcountries.eu (for country codes) and from our own XML4PharmaServer. Most of the latter will later be replaced by using the "CDISC Library API" RESTful web services. Remind that as regarding to CDISC standards, the "CDISC Library" is the "single source of truth".

Rules that use a RESTful web service have an indication by the metadata attribute ' webservice="…"'. For example:



```xml
<sdsrule id="SD2262" last-update="2019-08-14" originator="FDA" standard="SDTM" webservice="Uses NLM UNII Web Service">
<ruledescription>TSVAL/TSVALCD value must match for TRT</ruledescription>
<ruledetaileddescription>TSVAL and TSVALCD values must be populated from the same name record in FDA Substance Registration System (SRS)
<igversion>3.1.2</igversion>
<igversion>3.1.3</igversion>
<igversion>3.2</igversion>
<domain>TS</domain>
<rulexquery><![CDATA[
```

Rules not implemented

A number of the rules from the FDA Excel files have not been implemented, because:

- The rule is nonsense
- The rule is expected to lead to many "false positives"
- The rule is an expectation ("we would like to have it like this …") instead of a rule
- The rule is a duplication of another rule
- The rule cannot be technically implemented

The reason that there (unfortunately) appear such rules in the FDA publication, is probably due to that the FDA rules have been constructed based on "reviewers complaints", without good analysis, and without quality control by experts.

An example of such a "nonsense rule" is rule SD0029: "*Standard Units (--STRESU) should not be NULL, when Character Result/Finding in Std Units (--STRESC) is provided*".

We all know that there are so many tests for which the result does NOT have a unit. Examples are e.g. "pH", but also all tests for which the result is ordinal ("+1", "+2", or "positive", "negative"), nominal (e.g. the name of a species of bacteria) or narrative (free text).
Unfortunately, there are (too) many of such "nonsense" rules in the FDA publication.

Following is a list of the rules that have not been implemented:

Rule ID	**Rule description or FDA provided error message**	**Reason why the rule was not implemented**
SD1321	A treatment-emergent flag should be included in SUPPAE according to SDTM IG v3.1.2 #8.4.3	Duplicate of rule SD1097
SD2006	Unexpected MedDRA coding in the SUPPQUAL domain	Cannot be checked when the define.xml does not contain the information that it is a MedDRA code in SUPPQUAL
SD2259	TSVAL and TSVALCD values must be populated from the same concept description record in SNOMED CT	TODO: requires a RESTful web service to be available
SD0026	Original Units (--ORRESU) should not be NULL, when Result or Finding in Original Units (--ORRES) is provided	Nonsense rule. How can one be so stupid?
SD1082	Variable length should be assigned based on actual stored data to minimize file size. Datasets should be re-sized to the maximum length of actual data used prior to splitting	Does not apply to XML format. XML does not have a 200-character limitation
SD1096	High risk of truncated value for --TERM variable	"risk" has to do with probabilities. Unless one can quantify it, it does not belong into a rule

Rule ID	Rule description or FDA provided error message	Reason why the rule was not implemented
SD1116	Split datasets should have matching variable lengths for future merges. Datasets should be resized to the maximum length used prior to splitting	Does not apply to XML format – does not make sense when using XML
SD1117	The structure of Findings class domains should be one record per Finding Result per subject. No Finding Result with the same Test Short Name (--TESTCD) for the same Subject (USUBJID) and the same Collection Date (--DTC) are expected	This rule is nonsense. There are so many valid exceptions to this rule. Implementing it would automatically lead to a large number of false positives
SD1201	The structure of Events class domains should be one records per Event per subject. No Events with the same Collected Term (--TERM), Decoded Term (--DECOD), Category (--CAT), Subcategory (--SCAT), Severity (--SEV), and Toxicity Grade (--TOXGR) values for the same Subject (USUBJID) and the same Start Date (--STDTC) are expected	TO BE FURTHER investigated whether this can lead to false positives. Any way "are expected" should not appear in rules: rules must be exact
SD0007	Standard Units must be consistent within the same assessment (having the same --TESTCD, --CAT, --SCAT, --SPEC, --METHOD values)	This rule is nonsense. Simple example: "glucose in urine by dipstick method". All records have the same combination, but some results may be quantitative, others ordinal, depending on the lab that did the test.
SD0062	Study data must be provided in SAS XPORT v5 (.xpt) format	Is not implemented here as we use modern Dataset-XML format (a CDISC standard) instead of outdated XPT format
SD1324	Variable Label in define.xml must match variable Label in dataset	There are no variable labels in Dataset-XML format. Instead identifiers (OIDs) are used as is usual in modern formats
SD1325	Dataset Description in define.xml must match dataset Description	There are no dataset descriptions in Dataset-XML format. Instead identifiers (OIDs) are used as is usual in modern formats
SD0055	Variable Data Types in the dataset should match the variable data types described in SDTM	Will be implemented using the RESTful web services of the "CDISC Library API".
SD1076	Model permissible variable added into standard domain	This rule is nonsense: adding a model permissible variable is always allowed. If implemented, should never be more than an "info". Leads to many false positives in P21

Rule ID	**Rule description or FDA provided error message**	**Reason why the rule was not implemented**
CT2001	Variable must be populated with terms from its CDISC controlled terminology codelist. New terms cannot be added into non-extensible codelists.	TODO: second part requires RESTful web services from the CDISC Library API
CT2002	Variable should be populated with terms from its CDISC controlled terminology codelist. New terms can be added as long as they are not duplicates, synonyms or subsets of existing standard terms.	TODO: second part requires RESTful web services from the CDISC Library API. How can one check whether something is a synonym?
CT2004	Variable must be populated with terms from its CDISC controlled terminology codelist, when its value level condition is met. New terms cannot be added into non-extensible codelists	What is the difference with rule CT2001? It demonstrates lack of understanding (the role of) define.xml
CT2005	Variable should be populated with terms from its CDISC controlled terminology codelist, when its value level condition is met. New terms can be added as long as they are not duplicates, synonyms or subsets of existing standard terms	What is the difference with rule CT2002? It demonstrates lack of understanding (the role of) define.xml
CT2006	Paired variables such as TEST/TESTCD must be populated using terms with the same Codelist Code value in CDISC control terminology. There is one-to-one relationship between paired variable values defined in CDISC control terminology by Codelist Code value within the same value level condition	Duplicate of rule CT2003? What does it mean "paired … such as ..". The wording "such as" should not be in any rule: rules must be exact.
SD1073	Variables described in IG as inappropriate for usage must be not included in the dataset	The SDTM-IG does not describe any variables as "inappropriate", it only describes some as "not recommended". There are however so many examples of cases where such a variable is really needed. Implementation of this rule would either generate a large number of false positives, or encourage sponsors to "ban" such variables to SUPPQUAL
SD1120	Comments should be stored in Comments (CO) domain, rather than be put into Supplemental Qualifier (SUPP--) domains	This rule cannot be checked. As long as the FDA refuses to use CDISC-ODM for representing the (e)CRF, there is no machine-executable way to check this rule.

Rule ID	Rule description or FDA provided error message	Reason why the rule was not implemented
SD1119	In Comments (CO) domain, only following Identifier and Timing variables that are permissible and may be populated as appropriate when comments are not related to other domain records (RDOMAIN and IDVAR variables values are not populated): COGRPID, COREFID, COSPID, VISIT, VISITNUM, VISITDY, TAETORD, CODY, COTPT, COTPTNUM, COELTM, COTPTREF, CORFTDTC	TODO: currently it is unclear what this rule exactly means. To be discussed with the FDA.
SD0095	Supplemental Qualifiers special purpose dataset (SUPP--) can only be used to capture non-standard variables and their association to parent records in general-observation-class datasets (Events, Findings, Interventions) and Demographics	What does this rule mean? Does it mean that RDOMAIN can only be one of the ones from "Events", "Findings" or "Interventions", or "DM"? If so, the rule should say so. Rules must be clear!
SD2239	Planned Time Point Name (--TPT) value must be consistent for all records with same Subject (USUBJID) and Assessment Date/Time (--DTC)	This rule is nonsense. There can be several tests with different time points on the same date, even on the same datetime. The rule is not implemented as it may lead to many false positives.
SD1260	The Version of the Reference Terminology variable (TSVCDVER) should be populated with the version number of the Reference Terminology, when that Reference Terminology is versioned	How can it be known whether a terminology is versioned. Would have to be hardcoded in the rule for each possible terminology in the world.
SD2211	Diagnosis Group' (TDIGRP) record must be populated in Trial Summary (TS) domain, when study population is unhealthy subjects (HLTSUBJI = 'N')	Duplicate of rule SD1307
SD2223	Pharmacological Class of Investigational Therapy' (PCLAS) record must be populated in Trial Summary (TS) domain, when study type is 'INTERVENTIONAL' …	Duplicate of rule SD1312
SD2228	Intervention Model' (INTMODEL) record must be populated in Trial Summary (TS) domain, when study type is 'INTERVENTIONAL'	Duplicate of rule SD1310
SD2231	Intervention Type' (INTTYPE) record must be populated in Trial Summary (TS) domain, when study type is 'INTERVENTIONAL'	Duplicate of rule SD1311
SD0063 / SD0063A	Variable Label in the dataset should match the variable label described in SDTM IG	Duplicate of rule SD1324

Rule ID	Rule description or FDA provided error message	Reason why the rule was not implemented
SD9999	The structure for custom dataset should be based on one of the general observation classes (EVENTS, FINDINGS, INTERVENTIONS) defined by the SDTM model	It is not clear what this rule means. Is it about "one record per …"? Is it about the kind of variables present? Does it ignore the "def:Class" in the define.xml for custom domains?
SD1074	Variables designed only for SEND pre-clinical studies must be not included in the SDTM dataset	This rule needs further specification.
SD0029	Standard Units (--STRESU) should not be NULL, when Character Result/Finding in Std Units (--STRESC) is provided	Nonsense rule. How can one be so stupid?
SD2257 / SD2267 / SD2269	TSVAL for xxxxx record must be a valid term from SNOMED CT	TODO: requires SNOMED-CT RESTful web service. Could also be done from file, but would lead to version maintenance problems.
SD1290	If multiple disposition events are reported for an EPOCH, the sponsor should identify a primary reason and use that to populate DSTERM and DSDECOD. Additional reasons should be submitted in SUPPDS	It is unclear how this rule should be implemented. How can a system recognize that information in DSTERM and DSDECOD is a "reason"? Or was something else meant?

Other limitations

The NDF-RT coding system has been discontinued and replaced by MED-RT.
FDA has not reacted on this yet.
The RESTful web service for NDF-RT lookups has unfortunately also been decomissioned. See https://rxnav.nlm.nih.gov/MEDRT_transition.html for further details. It is also not possible to download recent NDF-RT files from the internet – these have all been removed.

As such, the rules that do a lookup in the NDF-RT system using the RESTful web service, will fail to execute and no messages will be produced when an invalid NDF-RT code is provided.