Paper SI05

Developing and Implementing CDISC CORE rules: from zero to hero

Marisa Wyckmans, SGS Health Science, Mechelen, Belgium
Els Janssens, SGS Health Science, Mechelen, Belgium

ABSTRACT

In clinical research, we have one mission: creating high quality data packages for statistical analysis, reporting, and ultimately submission to regulatory agencies.

Becoming a CDISC CORE volunteer was an exceptional opportunity to contribute to this. The enormous amount of conformance rules for SDTM, ADaM, and SEND makes this CDISC project a huge challenge. From project kick off with the volunteers in September 2021 until today, the CDISC CORE project evolved from basic concept into established validation system.

This paper will share experiences of a volunteer in the CDISC CORE Rules Development team. Examples will demonstrate the rule editor functionality and illustrate how easy developing your own rules can be, even for someone lacking solid programming skills.

In addition, time has now come to start implementation. Will we integrate the CDISC CORE engine in current systems or use it as a separate tool? And how about the company's own specific rules?

INTRODUCTION

Whether working at a pharmaceutical company, biotech or a CRO, everyone in clinical research has a global mission to bring effective medication to patients as fast as possible, and this is achieved by creating high quality data packages for statistical analysis, reporting, and ultimately submission to regulatory agencies.

Over the years, all possible aspects of a clinical trial have become more complex. There is for example more complexity in study designs, there are strict regulations and high quality of the data remains important. On top of all these data and documentation requirements, timelines are continuously challenged, requesting delivery of results as fast as possible.

At SGS, we believe that the CDISC CORE project, a free and open software allowing the entire clinical research industry to validate their own study data packages on conformance to CDISC foundational standards as well as to regulatory requirements, can help us to maintain high data quality and contribute to this global mission.

TO CORE OR NOT TO CORE

The call for volunteers in the CDISC CORE project was launched in the summer of 2021, and the project kicked off in September 2021. It was clear from the start that we, at SGS, wanted to contribute to this project.

AS A COMPANY

At SGS, we are already validating our data packages on an ongoing basis. In the future, CDISC CORE will bring a direct inflow of the CDISC Conformance rules into this validation. The CDISC Conformance rules are an integral part of the CDISC foundational standards and are always up to date with the latest standard, which means that CDISC CORE will allow us to unambiguously validate our data packages. In addition, regulatory rules will be part of CDISC CORE and even our own in-house developed rules can be added, bringing all validation rules together in one place. Moreover, CDISC CORE allows to only validate part of the data. This is especially interesting for SGS' blinded DM group who could, using CDISC CORE, validate only the subset of unblinding datasets they are responsible for.

As such, the open-source CORE software, is a way to achieve high quality in an efficient way.

AS A PERSON

As individuals, the CDISC CORE project allows us to expand our knowledge of the CDISC standards and to gain insight in how CDISC is defining and developing rules. Contributing to the CDISC CORE project allows to connect with people from the industry and is an exceptional way to contribute to the development of our industry.

PROFESSIONAL EXPERIENCE

With a background in Data Management, our main experience lies with reviewing EDC build, data cleaning, set-up and validation of SDTM, and defining and validating our data cleaning rules, which are SDTM based.

We have experience with SDTM and with rules in general, but are lacking solid programming skills and it would be our first-time volunteering for CDISC, so we were unsure at first: what is expected from us? Are we smart enough to do this? What if they only need programmers, and we are not the people they are looking for?

Nevertheless, we decided to sign up as volunteers for the CDISC CORE Rules Development Team.

PROJECT EVOLUTION

In the Rules Development Team, the project is built on 4 pillars:

- Training
- Team connection
- Information retrieval
- Process description and conventions

TRAINING

Training makes sure everyone understands the process and is repeated regularly for new people joining the Rules Development Team. From time to time, there are face-to-face workshops, for example at conferences.

TEAM CONNECTION

There are regular meetings for the Rules Development Team, in a very informal atmosphere, where everyone can ask questions, learn from each other, and help each other. These meetings are also a way to keep everyone informed on new decisions.

INFORMATION RETRIEVAL

When the project kicked off with the volunteers in September 2021, the relevant information was gathered on dedicated Wiki pages for the Rules Development Team. The most important pages were:

- Reference guide: The reference guide holds information about the different rule types, the possible operators to use and examples.
- SDTMIG Conformance Rules Sign-up List: The project kicked off with the SDTMIG 3.4 conformance rules, and the sign-up list was a list of all conformance rules for SDTMIG 3.4 to keep track of the status of each rule and who is working on it.

As the project evolved, the information from the reference guide was moved to GitHub, as GitHub allows better tracking of scripting updates. The sign-up list was replaced by a tracker in Jira, where every rule is listed as a Jira issue, which makes it easier to assign rules to people. In addition, a specific workflow is defined in Jira, allowing better follow-up of rule statusses.

Finally, there is a direct link between the Jira issue (a specific rule) and the GitHub issue (scripting update related to this specific rule).

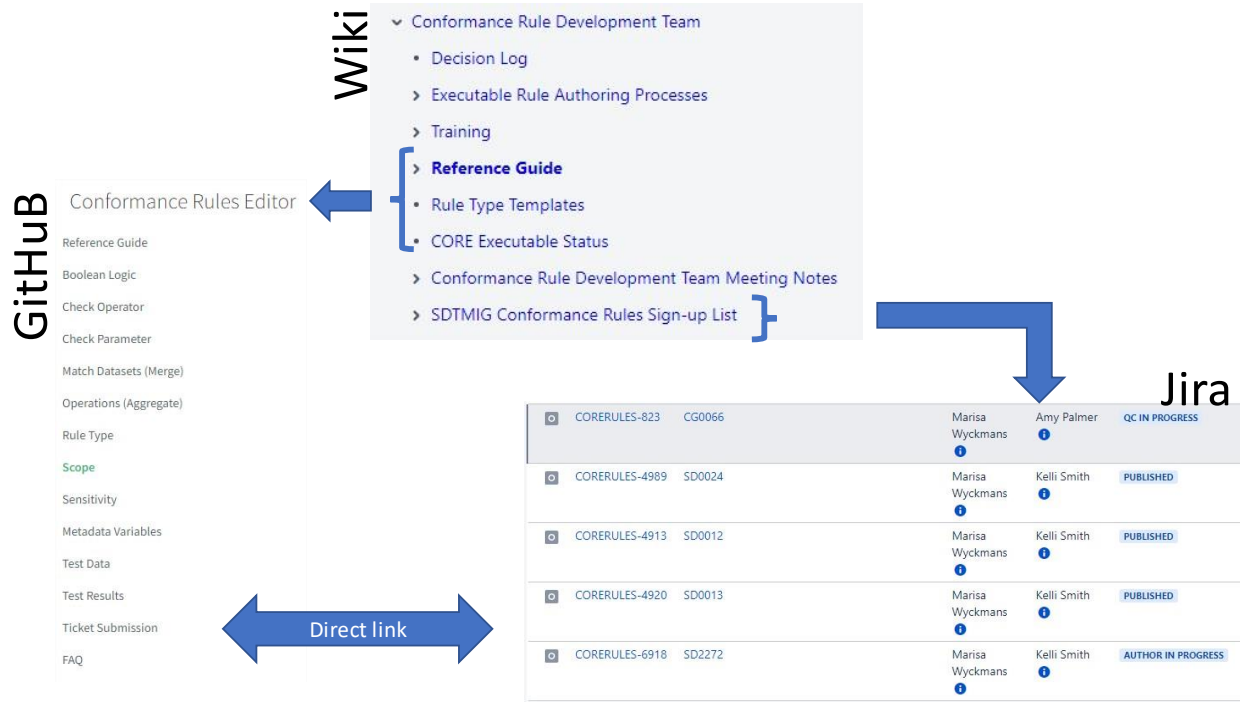


Figure 1: Evolution of information retrieval from Wiki to GitHub and Jira.

PROCESS DESCRIPTION AND CONVENTIONS

For the rule development process itself, proper documentation of the different steps in the process is needed. To ensure everyone (early volunteers versus new people joining later, small group versus big group) is taking the same steps, there is a fixed workflow that consists of 5 steps.

Figure 2: Fixed workflow for rule development, consisting of 5 steps.

- The first step is writing the rule in the rule editor, based on the source for the rule, which describes what the rule should do. The rule author will use the rule template, which is the backbone of mandatory components for the rule code, decide which operators to use, and develop the rule logic.
- The second step is creating test data for the rule, and this is also done by the rule author. The test data are created in an excel, and both positive and negative test data are required: positive means data are conformant and will not trigger output, negative means that data are not conformant and will trigger output. The test files are stored on a central SharePoint location, where they are available for the entire Rules Development Team.
- In the third step, a second person will take over and start the rule QC. During QC, the QC responsible will check that the rule in the rule editor matches the source, that the rule logic and syntax used are correct and that all components required by the schema are present.
- As soon as all technical aspects outlined in the previous step are considered ok, the rule output is validated by the same QC responsible who performed step 3. The test data created in step 2 is loaded in the system, the rule will run on the test data, and rule output is checked for correctness.
  In case of QC findings, there is a feedback loop to the rule author, from any of the QC steps. The QC findings are shared with the rule author, who can then make updates to the rule code and/or test data, and QC will be repeated afterwards. If the rule output is considered ok, the results are saved on the SharePoint together with the test data to prove successful validation of the rule.
- When QC is finished without any findings, the last step is publishing the rule in the CDISC library.

HARDCORE PRACTICE

The example below illustrates how rule development works in practice.

STEP 1 – WRITE RULE

The source for this example is the SDTM and SDTMIG Conformance Rules v2.0 excel file:

Rule ID	SDTMIG Version	Rule Version	Class	Domain	Variable	Condition	Rule
CG0066	3.2	1	EVT	DS	DSDECOD	DSCAT = 'PROTOCOL MILESTONE'	DSTERM = DSDECOD

Figure 3: Snapshot from the SDTM and SDTMIG Conformance rules v2.0 for rule CG0066.

Following information is available in the source:

- Rule identifiers: The Rule ID, SDTMIG Version, and Rule Version
- Scope: The Class, Domain and Variable the rule applies to
- Rule details: the Rule, and the Condition

The information from the source file is translated into rule code in the rule editor using simple YAML syntax and the rules are written to give output in case data is not conformant. Following components are part of the rule code:

- Rule logic (or Check, see lines 42-50 in Figure 4)
  - The rule definition in the source file says that DSTERM must equal DSDECOD, meaning the rule should output the cases where DSTERM is not equal to DSDECOD, and this is put in the rule logic (see lines 48-50 in Figure 4).
  - The rule also has a condition, it only applies when DSCAT is equal to 'PROTOCOL MILESTONE'. The condition should be part of the rule logic as well to ensure that the rule is limited to data in scope, i.e., the records where DSCAT is equal to 'PROTOCOL MILESTONE' (see lines 44-47 in Figure 4).

- Description (see line 26 in Figure 4): the description describes the purpose of the check.
- Outcome (see lines 27-31 in Figure 4):
    - Message (see line 28 in Figure 4): The message will be visible in the output of the rule and describes what is wrong in the data.
    - Output variables (see lines 29-31 in Figure 4): The variables listed here will be shown in the rule output. Since this rule is comparing DSTERM and DSDECOD, these variables are chosen to be part of the output.
- Authorities (see lines 4-21 in Figure 4): The rule identifier information from the source is also incorporated in the rule editor to ensure the rule code of every rule mentions the standard and version the rule is applicable to and has a reference to the source document.
- Scope (see lines 34-40 in Figure 4): The class, and domain in scope of the rule are included as well.

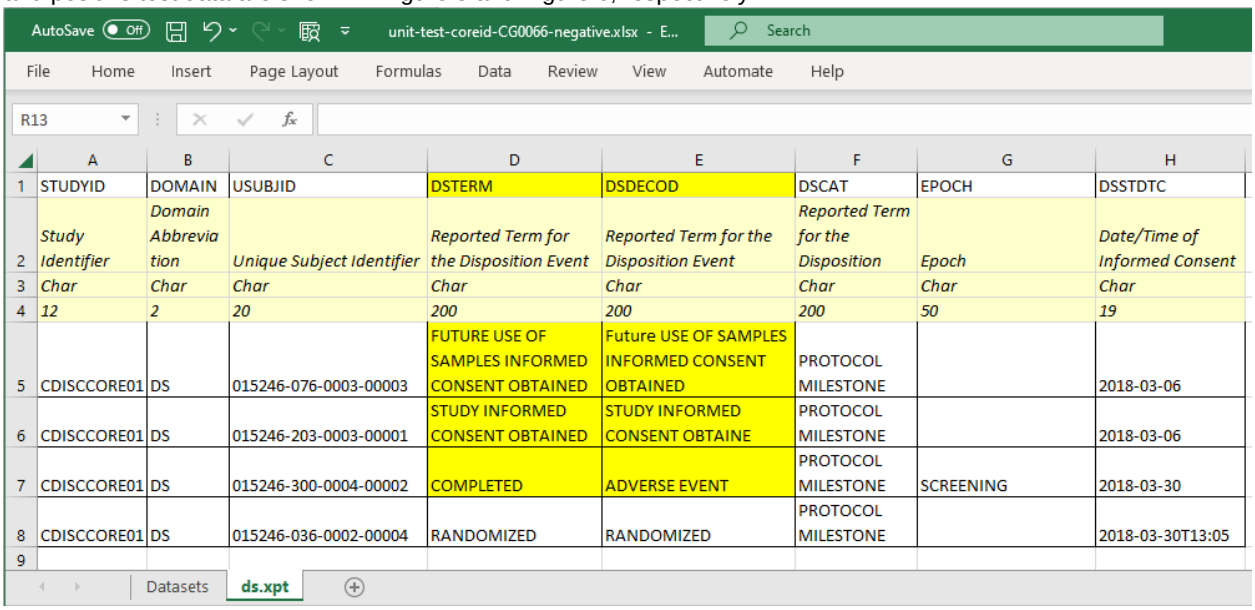
EDIT TEST DIFF

```yaml
# Variable: DSDECOD
# Condition: DSCAT = 'PROTOCOL MILESTONE'
# Rule: DSTERM = DSDECOD
Authorities:
  - Organization: CDISC
    Standards:
      - Name: SDTMIG
        Version: '3.2'
        References:
          - Origin: SDTM and SDTMIG Conformance Rules
            Rule Identifier:
              Id: CG0066
              Version: '1'
            Version: '2.0'
            Citations:
              - Cited Guidance: d. When DSCAT='PROTOCOL MILESTONE', DSTERM and DSDECOD will
                  contain the same value drawn from the sponsor's controlled
                  terminology.
                Document: IG v3.2
                Item: Assumption 3d
                Section: '6.2'
Core:
  Id: CORE-000322
  Version: '1'
  Status: Published
Description: 'When DSCAT equals PROTOCOL MILESTONE, then DSTERM should be equal to DSDECOD'
Outcome:
  Message: 'DSCAT equals PROTOCOL MILESTONE but DSTERM is not equal to DSDECOD'
  Output Variables:
    - DSTERM
    - DSDECOD
Rule Type: Record Data
Sensitivity: Record
Scope:
  Classes:
    Include:
      - EVENTS
  Domains:
    Include:
      - DS
Executability: Fully Executable
Check:
  all:
    - name: DSCAT
      operator: equal_to
      value: PROTOCOL MILESTONE
      value_is_literal: true
    - name: DSTERM
      operator: not_equal_to
      value: DSDECOD
```

Figure 4: Rule CG0066 in the editing module of the Rule Editor.

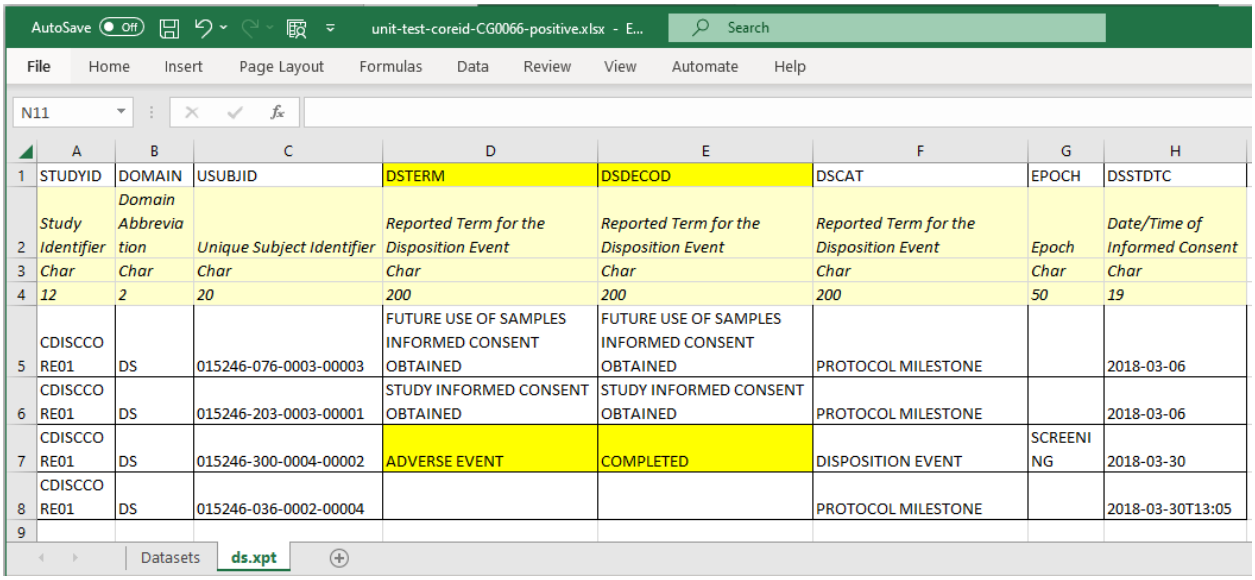
STEP 2 – CREATE TEST DATA

Test data needs to be created for each rule considering that there should be positive (does not trigger output) and negative (triggers output) test data, and there should be sufficient test data to cover each aspect of the rule. Negative and positive test data are shown in Figure 5 and Figure 6, respectively.



	A	B	C	D	E	F	G	H
1	STUDYID	DOMAIN	USUBJID	DSTERM	DSDECOD	DSCAT	EPOCH	DSSTDTC
2	*Study Identifier*	*Domain Abbreviation*	*Unique Subject Identifier*	*Reported Term for the Disposition Event*	*Reported Term for the Disposition Event*	*Reported Term for the Disposition*	*Epoch*	*Date/Time of Informed Consent*
3	*Char*	*Char*	*Char*	*Char*	*Char*	*Char*	*Char*	*Char*
4	*12*	*2*	*20*	*200*	*200*	*200*	*50*	*19*
5	CDISCCORE01	DS	015246-076-0003-00003	FUTURE USE OF SAMPLES INFORMED CONSENT OBTAINED	Future USE OF SAMPLES INFORMED CONSENT OBTAINED	PROTOCOL MILESTONE		2018-03-06
6	CDISCCORE01	DS	015246-203-0003-00001	STUDY INFORMED CONSENT OBTAINED	STUDY INFORMED CONSENT OBTAINE	PROTOCOL MILESTONE		2018-03-06
7	CDISCCORE01	DS	015246-300-0004-00002	COMPLETED	ADVERSE EVENT	PROTOCOL MILESTONE	SCREENING	2018-03-30
8	CDISCCORE01	DS	015246-036-0002-00004	RANDOMIZED	RANDOMIZED	PROTOCOL MILESTONE		2018-03-30T13:05
9								

Figure 5: Excel file containing negative test data for CG0066. The yellow highlighted records are expected to trigger output as DSCAT = PROTOCOL MILESTONE and DSTERM is not equal to DSDECOD.



	A	B	C	D	E	F	G	H
1	STUDYID	DOMAIN	USUBJID	DSTERM	DSDECOD	DSCAT	EPOCH	DSSTDTC
2	*Study Identifier*	*Domain Abbreviation*	*Unique Subject Identifier*	*Reported Term for the Disposition Event*	*Reported Term for the Disposition Event*	*Reported Term for the Disposition Event*	*Epoch*	*Date/Time of Informed Consent*
3	*Char*	*Char*	*Char*	*Char*	*Char*	*Char*	*Char*	*Char*
4	*12*	*2*	*20*	*200*	*200*	*200*	*50*	*19*
5	CDISCCORE01	DS	015246-076-0003-00003	FUTURE USE OF SAMPLES INFORMED CONSENT OBTAINED	FUTURE USE OF SAMPLES INFORMED CONSENT OBTAINED	PROTOCOL MILESTONE		2018-03-06
6	CDISCCORE01	DS	015246-203-0003-00001	STUDY INFORMED CONSENT OBTAINED	STUDY INFORMED CONSENT OBTAINED	PROTOCOL MILESTONE		2018-03-06
7	CDISCCORE01	DS	015246-300-0004-00002	ADVERSE EVENT	COMPLETED	DISPOSITION EVENT	SCREENING	2018-03-30
8	CDISCCORE01	DS	015246-036-0002-00004			PROTOCOL MILESTONE		2018-03-30T13:05
9								

Figure 6: Excel file containing positive test data for CG0066. These records are not expected to trigger any output: DSTERM equals DSDECOD for the 3 records with DSCAT = PROTOCOL MILESTONE; for the yellow highlighted record, DSTERM is not equal to DSDECOD, however, DSCAT = DISPOSITION EVENT and the rule only applies to DSCAT = PROTOCOL MILESTONE.

STEP 3 – QC RULE DESCRIPTION

When the rule is developed in the rule editor, and test data is created and available on SharePoint, a second person can start the validation of the rule. The QC responsible will check the source (Figure 3) and the rule code in the rule editor (Figure 4) to ensure that the information in the rule code matches the source. Next, 3 technical aspects of the rule are checked automatically in the TEST module of the rule editor (Figure 7). The system will validate the YAML syntax and the schema and check whether the YAML code from the rule editor can be successfully converted to a JSON rule.

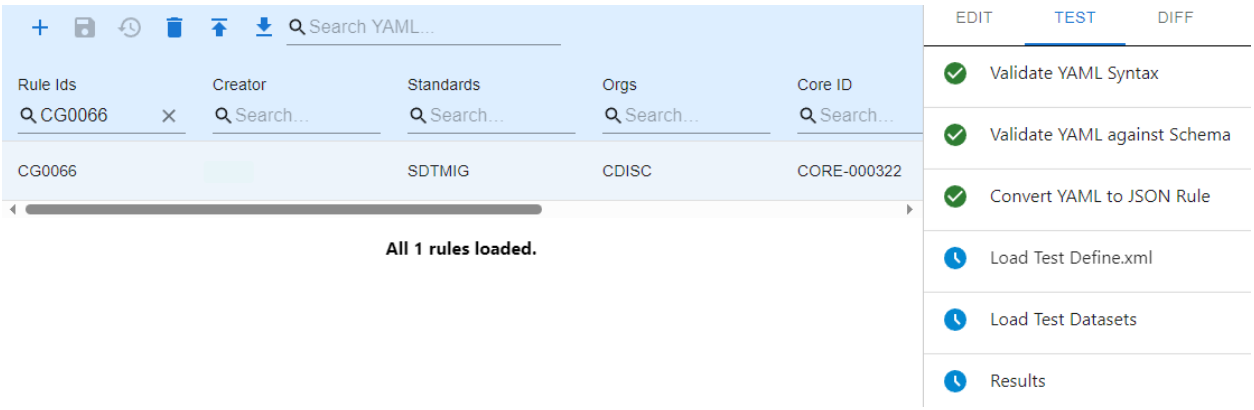


Figure 7: Rule CG0066 in the testing module of the Rule Editor. The 3 green checkmarks demonstrate successful completion of the 3 technical aspects of the rule validation.

STEP 4 – VALIDATE RULE OUTPUT

The next step is loading the test data. The QC responsible choses the correct file from SharePoint and the system will load the test data.

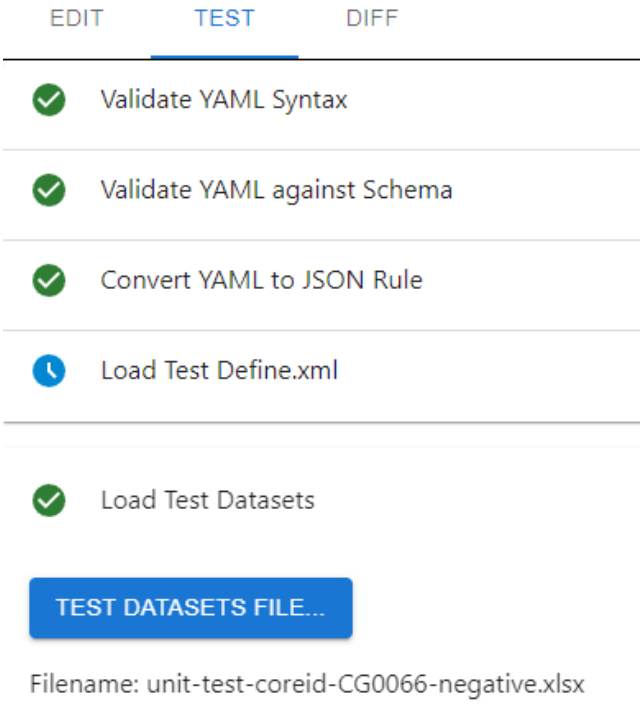


Figure 8: Rule CG0066 in the testing module of the Rule Editor, with successful upload of the negative test data.

When the upload of test data is successful, the green checkmark appears for this step and the system shows the filename of the file that was loaded. Figure 8 illustrates the successful upload of the negative test data for CG0066 (Figure 5 shows the content of these negative test data).

While loading the test data, the system will also automatically run the rule on these test data and the results are immediately visible in the testing module of the Rule Editor. In the negative test data for CG0066, 3 records are expected to trigger output (see yellow highlighted records in Figure 5).

The QC responsible will check that the results in the testing module of the Rule Editor are as expected (Figure 9): there is a green checkmark for Results, and “executionstatus” = “success”, the summary in the system mentions there are 3 negatives which is as expected based on our test data. As a final check during the validation, the QC responsible will check the result details to ensure the rule output matches the records that are expected to trigger output. The result details (i.e., values of DSDECOD, DSTERM and USUBJID that are shown under “errors”) in Figure 9 match the 3 records that are expected to trigger output for CG0066 (Figure 5).

EDIT TEST DIFF

Validate YAML Syntax

Validate YAML against Schema

Convert YAML to JSON Rule

Load Test Define.xml

Load Test Datasets

Results Negatives 3

Results


```
{ 1 item
  "DS" : [ 1 item
    0 : { 5 items
      "executionStatus" : "success"
      "domain" : "DS"
      "variables" : [ 2 items
        0 : "DSDECOD"
        1 : "DSTERM"
      ]
      "message" : "DSCAT equals PROTOCOL MILESTONE but DSTERM is not equal to DSDECOD"
      "errors" : [ 3 items
        0 : { 3 items
          "value" : { 2 items
            "DSTERM" : "FUTURE USE OF SAMPLES INFORMED CONSENT OBTAINED"
            "DSDECOD" : "Future USE OF SAMPLES INFORMED CONSENT OBTAINED"
          }
          "row" : 1
          "USUBJID" : "015246-076-0003-00003"
        }
        1 : { 3 items
          "value" : { 2 items
            "DSTERM" : "STUDY INFORMED CONSENT OBTAINED"
            "DSDECOD" : "STUDY INFORMED CONSENT OBTAINE"
          }
          "row" : 2
          "USUBJID" : "015246-203-0003-00001"
        }
        2 : { 3 items
          "value" : { 2 items
            "DSTERM" : "COMPLETED"
            "DSDECOD" : "ADVERSE EVENT"
          }
          "row" : 3
          "USUBJID" : "015246-300-0004-00002"
        }
      ]
    }
  ]
}
```


Figure 9: Rule CG0066 in the testing module of the Rule Editor, demonstrating successful rule execution and showing rule output results.

These steps (loading test data and checking results) must be repeated for the positive test data.
Only if validation with both positive and negative test data is successful and results meet the expectations, the rule is ready to publish.

FUTURE DEVELOPMENT

While rule development is further progressing, it is time to start thinking about implementing CDISC CORE. Can the CDISC CORE engine be integrated in our current systems, or should it be used as a separate tool?
Currently, the CORE engine is integrated in our daily conversion process, which allows the currently available rules to run in the background, and to evaluate all technical aspects, performance, and system load.
When a complete CDISC CORE rule package is available, we will be ready to use CDISC CORE in production and perform validation of our data packages via CORE. Since it is easily integrated in our daily conversion process, outputs will be refreshed daily.
Next an exploratory initiative was started to deploy a local installation of the rule editor in the SGS cloud to create and run custom rules.

CONCLUSION

Rule writing and validation is very doable and very accessible for everyone. The only must-have is knowledge of the CDISC standard you work on. The reason why it is very accessible for everyone is the amazing work behind the scenes to turn the simple YAML code into complex scripts, but let's also not forget the hard work from project leads at CDISC and volunteers to get where rule development is now.
Of course, not everything was always smooth. It is impossible to foresee everything upfront in such a huge project and especially in the beginning, some rework on the rules was needed as development of the rule editor and schema continued while rule authoring was ongoing. To avoid confusion, all information should be kept up-to-date and obsolete information should be removed. There are non-executable or ambiguous rules, so not everything can be made machine-readable. With a diverse group of people working on the rules, there is also a challenge in keeping everything aligned.
Becoming a volunteer for the CDISC CORE Rules Development Team in 2021 was the right decision. Over the last 2 years, we authored, QCed and published rules, even without solid programming skills. Exciting times are still ahead of us, both for rule development as well as implementation. Being an active member of the Rules Development Team is a fun and useful way to contribute to our industry. Will you join?

REFERENCES

https://www.cdisc.org/core

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Marisa Wyckmans
Els Janssens
SGS Health Science
Generaal De Wittelaan 19A bus 5
BE - 2800 Mechelen
Work Phone:
EUROPE: +32 15 27 32 45
AMERICAS: + 1 877 677 2667
Email: clinicalresearch@sgs.com
Web: www.sgs.com/cro