CDISC CORE

31st German CDISC User Network Meeting, Munich, 2024-03-14

Jozef Aerts
XML4Pharma

What is CDISC CORE?

- CORE = CDISC Open Rules Engine

- Meant to replace the "jungle" of validation rules and implementations

- Open source, vendor-neutral, extensible

- All implementations should give the same results

CDISC CORE and FDA

- FDA has asked CDISC to implement the "FDA Business Rules" into CORE
  https://www.cdisc.org/news/cdisc-proud-announce-research-collaboration-incorporate-fda-business-rules-cdiscs-open-rules

- It is expected that FDA will replace P21 by CORE this year or next year

Who is developing the CORE rules?

- CORE rules are developed starting from the existing CDISC validation rules (mostly as Excel) by a large number of volunteers (>100)

- Engine programming in Python is done by 2+ specialists at CDISC

- Everything is published as "open source" on Github: https://github.com/cdisc-org/cdisc-rules-engine/

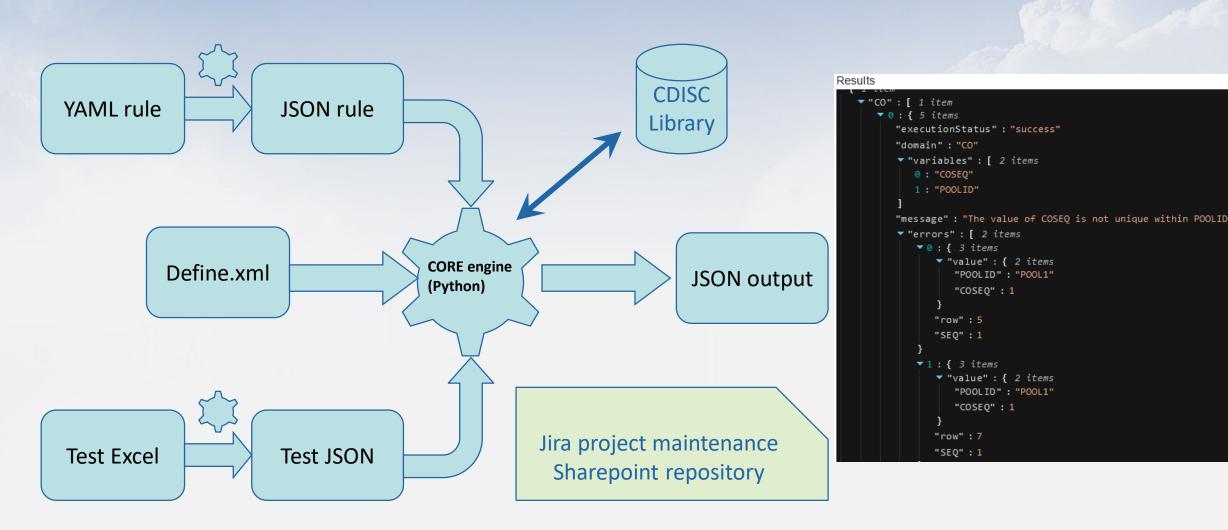# Rule implementation made easy by YAML

YAML Rule editor used by the volunteers

Test files as Excel

How does that work? Development

How does that work? Production
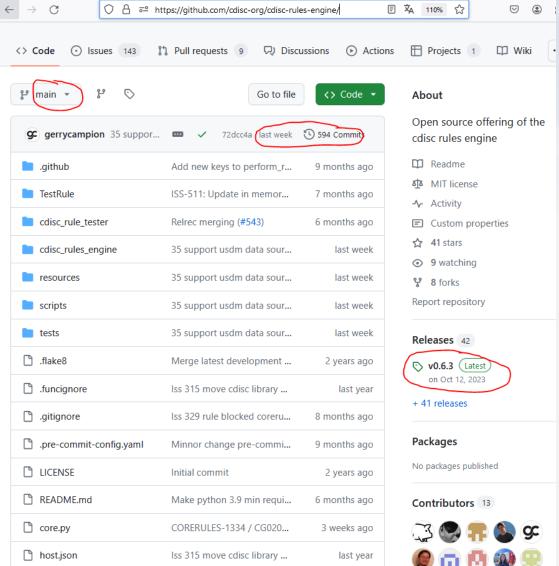
How do I get it?

- Download from Github
  - source code
    https://github.com/cdisc-org/cdisc-rules-engine/

  - Windows executable

- Use vendor implementation
  - Formedix CORE (outdated)
  - Smart Submission Dataset Viewer
  - SDTM-ETL mapping software

Jozef Aerts, German UG CDISC M

How do I use it?

- "Quick start" on Github
- Essentially using CLI (Command Line Interface)
- Commands are valid for both Python as Windows-exe implementation

- Also instructions how to compile (latest version) to Windows exe

Running a validation

From the command line

Clone the repository and run `python core.py --help` to see the full list of commands.

Run `python core.py validate --help` to see the list of validation options.

```
-ca, --cache TEXT            Relative path to cache files containing pre
                             loaded metadata and rules
-ps, --pool-size INTEGER      Number of parallel processes for validation
-d, --data TEXT              Path to directory containing data files
-dp, --dataset-path TEXT     Absolute path to dataset file. Can be specified
-dxp, --define_xml_path TEXT  Path to Define-XML
-l, --log-level [info|debug|error|critical|disabled|warn]
                             Sets log level for engine logs, logs are
                             disabled by default
-rt, --report-template TEXT  File path of report template to use for
                             excel output
-s, --standard TEXT          CDISC standard to validate against
                             [required]
-v, --version TEXT           Standard version to validate against
                             [required]
-ct, --controlled-terminology-package TEXT
                             Controlled terminology package to validate
                             against, can provide more than one
-o, --output TEXT            Report output file destination
-of, --output-format [JSON|XLSX]
                             Output file format
-rr, --raw-report            Report in a raw format as it is generated by
                             the engine. This flag must be used only with
                             --output-format JSON.
-dv, --define-version TEXT   Define-XML version used for validation
-dxp, --define-xml-path      Path to define-xml file.
```

How do I use it? List of Rules

- list rules:

  - list-rules - list rules available in the cache

    - list all rules:

      `python core.py list-rules`

    - list rules for standard:

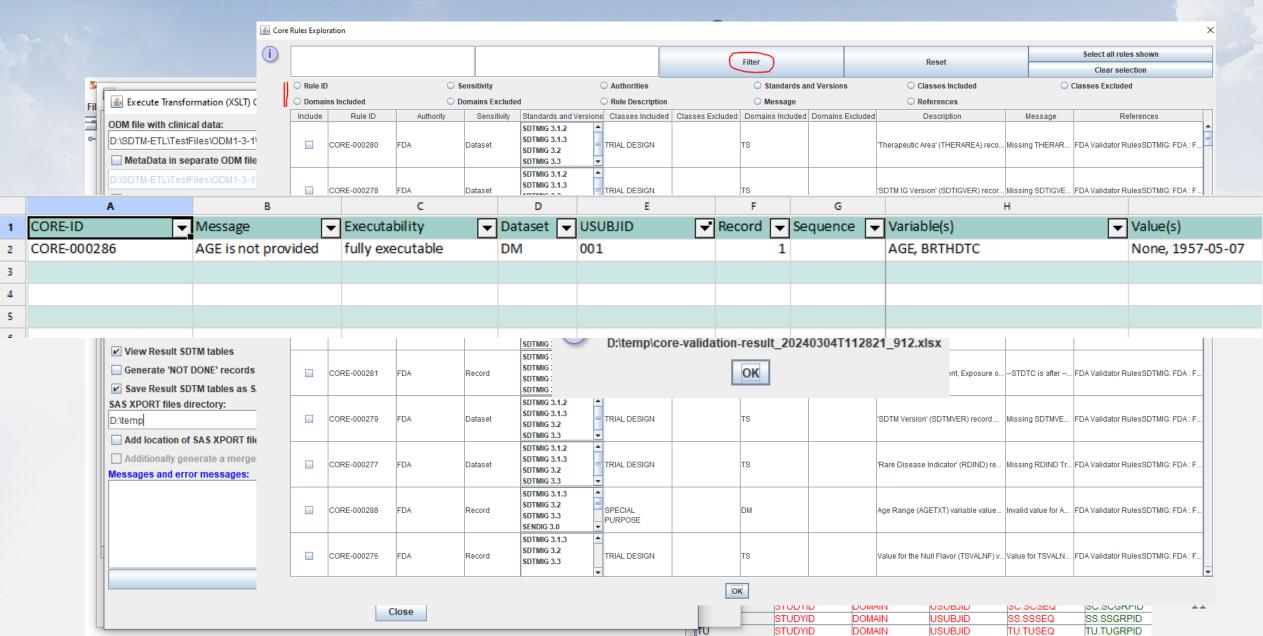   `python core.py list-rules -s sdtmig -v 3-4`

  - list-rule-sets - lists all standards and versions for which rules are available: `python core.py list-rule-sets`

- update rules (requires CDISC-Library API key):

  - update-cache - update locally stored cache data (Requires an environment variable - `CDISC_LIBRARY_API_KEY`)

    `python core.py update-cache` ←

    To obtain an api key, please follow the instructions found here: https://wiki.cdisc.org/display/LIBSUPRT/ Getting+Started%3A+Access+to+CDISC+Library+API+using+API+Key+Authentication. Please note it can take up to an hour after sign up to have an api key issued

Implementation in SDTM-ETL

Demo time!

Using CORE in the open source Smart Submission Dataset Viewer