FDA business rules and CDISC Open Rules, the road to adoption

Nick De Donder, CDISC Open Rules Program Manager, CDISC
Jozef Aerts, Owner, XML4Pharma

Meet the Speakers

Nick De Donder

Title: CDISC Open Rules Program Manager

Organization: CDISC

Nick has a diverse work experience in the field of life sciences and data standards. Nick started working at Business & Decision Life Sciences (now Enovalife) in 2008, where he held various roles. Recently he became an independent consultant. Since June 2021 Nick has been CDISC Open Rules Program Manager. Since 2017, Nick has been a CDISC Authorized Trainer and progressively took on roles such as E3C member, E3C Chair, and PHUSE Committee member.

Jozef Aerts

Title: Owner

Organization: XML4Pharma

Passionate about standards in clinical research and healthcare, and their implementation in IT systems.

Disclaimer and Disclosures

- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of CDISC.*

Agenda

1. Research Collaboration Agreement
2. FDA Business Rules
3. Into Practice
4. Technical Implementation of CORE
5. Conclusions

FDA Business Rules

Background

- Sponsors should evaluate their study data before submission against the conformance rules published by an SDO, the eCTD Technical Rejection Criteria for Study Data, and the FDA Business Rules.
- The Business Rules v1.5 (May 2019) help ensure that the study data are compliant, useful, and will support meaningful review and analysis. This applies to SDTM formatted clinical studies and SEND formatted non-clinical studies. For more information, see Section 8 of the Technical Conformance Guide.
- All business rules should be followed where applicable.

Research Collaboration Agreement with US FDA CDER & CBER

CDISC and FDA working together to develop machine-executable formats of the FDA Business Rules and on the development and ongoing governance of this set of executable rules within the CDISC Open Rules project that can be used by industry

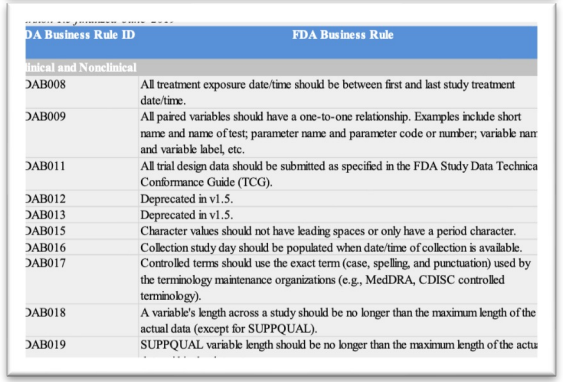
The benefits of creating a single source of truth for all FDA validation needs increases transparency for all stakeholders on how FDA checks data for conformance to CDISC standards and to existing FDA Business Rules

CDISC is Proud to Announce a Research Collaboration to Incorporate FDA Business Rules into CORE

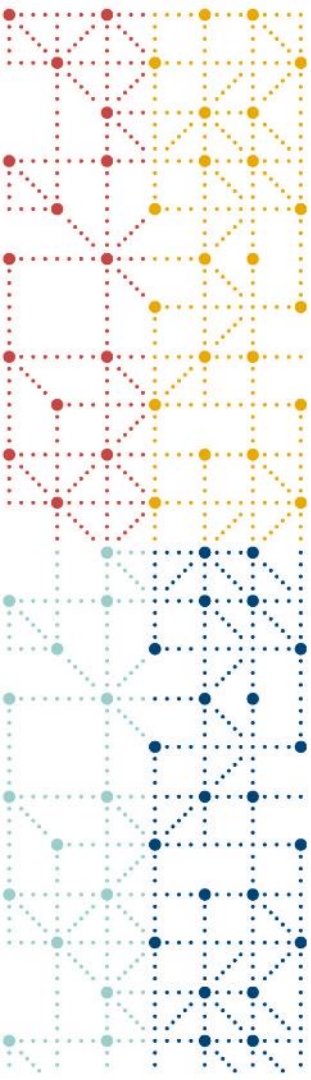
ustin, TX – January 16, 2024 – CDISC is proud to announce a research collaboration with he U.S. Food and Drug Administration's Office of Translational Sciences in the Center for Drug Evaluation and Research and Office of Regulatory Operations in the Center for Biologics Evaluation and Research to incorporate FDA Business Rules into CDISC's Open Rules Engine (CORE).

CDISC's CORE project provides an open-source version of the CDISC Conformance Rules n a machine-executable format. These rules, published and managed by CDISC, create a single source for conformance rules and allow external vendors and sponsor companies to mplement and extend these rules within their tools. FDA Business Rules are currently vritten in a plain text, non-machine executable format and describe the business equirements for regulatory review to help ensure that clinical trial study data is compliant nd useful and supports meaningful review and analysis.

The goal of this effort, which began on November 3, 2023 and has term of three years, is to collaborate on providing input on machine-executable formats of the FDA Business Rules nd on the development and ongoing governance of this set of executable rules within CORE



DA Business Rule ID	FDA Business Rule
linical and Nonclinical	
DAB008	All treatment exposure date/time should be between first and last study treatment date/time.
DAB009	All paired variables should have a one-to-one relationship. Examples include short name and name of test; parameter name and parameter code or number; variable nam and variable label, etc.
DAB011	All trial design data should be submitted as specified in the FDA Study Data Technica Conformance Guide (TCG).
DAB012	Deprecated in v1.5.
DAB013	Deprecated in v1.5.
DAB015	Character values should not have leading spaces or only have a period character.
DAB016	Collection study day should be populated when date/time of collection is available.
DAB017	Controlled terms should use the exact term (case, spelling, and punctuation) used by the terminology maintenance organizations (e.g., MedDRA, CDISC controlled terminology).
DAB018	A variable's length across a study should be no longer than the maximum length of the actual data (except for SUPPQUAL).
DAB019	SUPPQUAL variable length should be no longer than the maximum length of the actu



Into Practice

FDA Business Rules

Version 1.5 finalized June 2019	
FDA Business Rule ID	**FDA Business Rule**
Clinical and Nonclinical (SDTMIG, SEND)	
FDAB008	All treatment exposure date/time should be between first and last study treatment date/time.
FDAB009	All paired variables should have a one-to-one relationship. Examples include short name and name of test; parameter name and parameter code or number; variable name and variable label, etc.
FDAB011	All trial design data should be submitted as specified in the FDA Study Data Technical Conformance Guide (TCG).
FDAB015	Character values should not have leading spaces or only have a period character.
FDAB016	Collection study day should be populated when date/time of collection is available.
FDAB017	Controlled terms should use the exact term (case, spelling, and punctuation) used by the terminology maintenance organizations (e.g., MedDRA, CDISC controlled terminology).
FDAB018	A variable's length across a study should be no longer than the maximum length of the actual data (except for SUPPQUAL).
FDAB019	SUPPQUAL variable length should be no longer than the maximum length of the actual data within the dataset.
FDAB024	Large datasets should be split into smaller datasets no larger than 5 GB in size.
FDAB026	Records with a baseline flag should have a corresponding standard result with a standardized unit where available.
FDAB030	Standard units should be consistent within the same assessment (having the same --TESTCD, --CAT, --SCAT, --SPEC, --METHOD values).

Identifiers				Scope of Rule					
Rule ID	Rule ID Version (represents any change to the rule)	Related Rule(s) [See Also, Compare Against]	Rule Set (Generally IG Version, OCCDS v1.0, ADNCA v1	Class	Subclass	Dataset or Domai	Variable	Elemen	Scope Sectio
FB0801	1			INT		EC, EX (merged w/DM)	--STDTC, --ENDTC, RFXSTDTC, RFXENDTC	ItemGroup Def ItemDef	
FB0901	1		All SDTMIG, SENDIG	ALL		ALL	Variable name and variable label		
FB0902	1		All SDTMIG, SENDIG	FND, TDM		All	--TESTCD, --TEST		
FB0903	1		All SDTMIG, SENDIG	SPC, TDM		DM, TA, TV, TE	DM.ARM-DM.ARMCD, DM.ACTARM-DM.ACTARMCD, TA.ARM-TA.ARMCD, TA.ACTARM-TA.ACTARMCD, TV.ARM-TV.ARMCD, TV.ACTARM-TV.ACTARMCD, TE.ELEMENT-TE.ETCD		
FB0904	1		All SDTMIG, SENDIG	TDM		TS	TS.TSVAL-TS.TSVALCD-TS.TSVALNF		
FB0905	1		All SDTMIG	INT		AG, CM, SU	--CLAS, --CLASCD		
FB0910	1		All SDTMIG	TDM		TI	TI.IETEST, TI.IETESTCD		

Harmonized spreadsheet

Identifiers							Release Notes		
Rule ID	Rule ID Version (represents any change to the rule)	Related Rule(s) [See Also, Compare Against	Rule Set (Generally IG Version, OCCDS v1.0, ADNCA v1	Scope Sectio	Rule Sectic	Standards Secti	Guidance Section	Release Notes	Authoring sectior

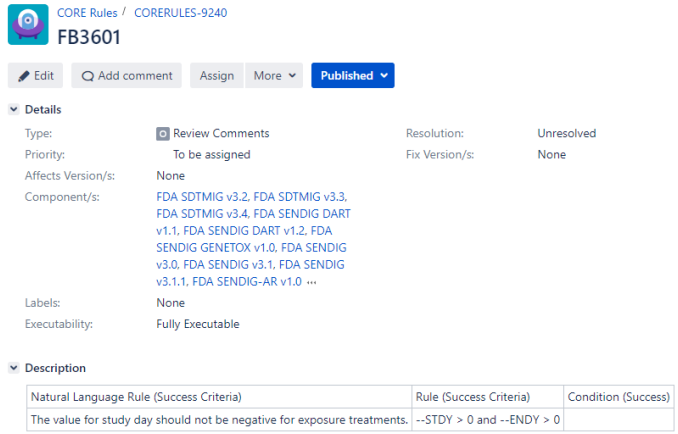
Business specification

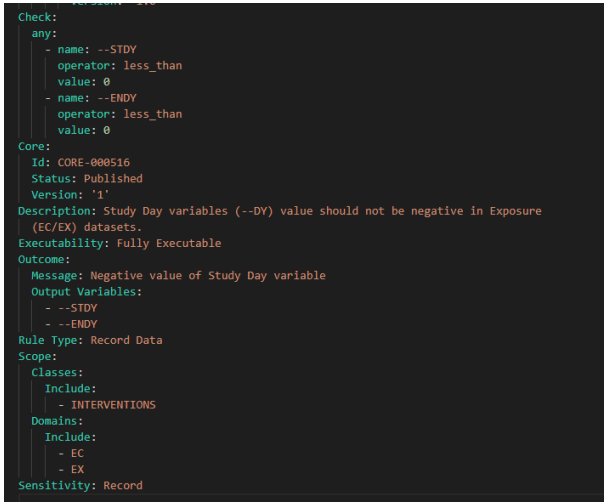
Identifiers						Statement of Rule						
Rule ID	Rule ID Version (represents any change to the rule)	Related Rule(s) (See Also, Compare Against	Rule Set (Generally IG Version, OCCDS v1.0, ADNCA v1	Scope Sectio	Natural Language Rule (Success Criteria)	Rule (Success Criteria)	Condition (Success)	Natural Language Rule (Failure Criteria)	Rule (Failure Criteria)	Condition (Failure)	Error Message	Executabil
FB0801	1				All treatment exposure date/time should be between first and last study treatment date/time.	(RFXSTDTC <= --STDTC <= --ENDTC <= RFXENDTC	Variables exists and are populated; otherwise, account for all possibilities and treat with ANY conditions. Note: need functionality 'check for RFX--DTC availability in IG' for generic rule implementation. Note: NA for earlier SENDIGs that don't have RFX--DTC.	All treatment exposure date/time outside first and last study treatment date/time.	not (RFXSTDTC <= --STDTC <= --ENDTC <= RFXENDTC)	Variables exists and populated; otherwise, account for all possibilities and treat with ANY conditions.	All treatment exposure date/time outside first and last study treatment date/time.	Fully executable
FB0901	1		All SDTMIG, SENDIG			One-to-one		One-to-one mapping is not maintained.	Variable name and label pairing is not unique		One-to-one mapping is not maintained.	Fully executable
FB0902	1		All SDTMIG, SENDIG		TESTCD and TEST pair should have one-to-one relationship	One-to-one		One-to-one mapping is not maintained.	Pairing is not unique		One-to-one mapping is not maintained.	Fully executable
FB0903	1		All SDTMIG, SENDIG			One-to-one		One-to-one mapping is not maintained.	Pairing is not unique		One-to-one mapping is not maintained.	Fully executable
FB0904	1		All SDTMIG, SENDIG			One-to-one	TSVALCD ^= null	One-to-one mapping is not maintained.	Pairing is not unique		One-to-one mapping is not maintained.	Fully executable
FB0905	1		All SDTMIG			One-to-one		One-to-one mapping is not maintained.	Pairing is not unique		One-to-one mapping is not maintained.	Fully executable
FB0910	1		All SDTMIG			One-to-one		One-to-one mapping is not maintained.	Pairing is not unique		One-to-one mapping is not maintained.	Fully executable

Business rule to executability

FDAB036	The value for study day should not be negative for exposure treatments.

Identifiers					Statement of Rule							
Rule ID	Rule ID Version (represents any change to the rule)	Related Rule(s) (See Also, Compare Against	Rule Set (Generally IG Version, OCCDS v1.0, ADNCA v1	Scope Sectio	Natural Language Rule (Success Criteria)	Rule (Success Criteria)	Condition (Success)	Natural Language Rule (Failure Criteria)	Rule (Failure Criteria)	Condition (Failure)	Error Message	Executabili
FB3601	1				The value for study day should not be negative for exposure treatments.	--STDY > 0 and --ENDY > 0		The value for study day is negative for exposure treatments.	--STDY < 0 or --ENDY < 0			Fully executable





Test data

- Positive test data does not show issues

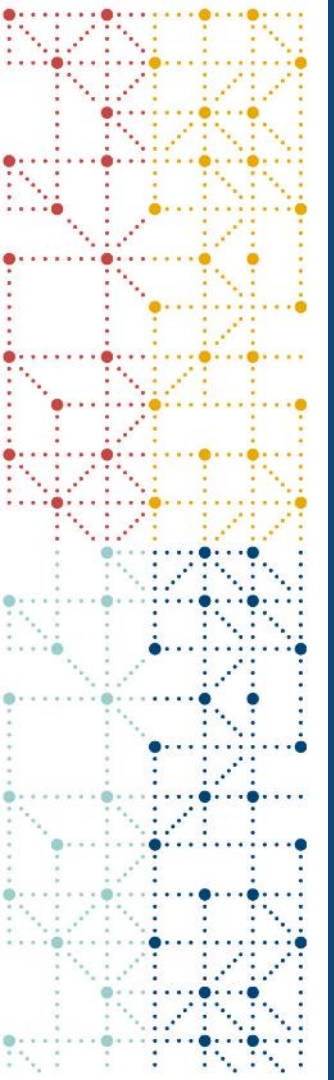
EPOCH	EXSTDTC	EXENDTC	EXSTDY	EXENDY
Epoch	*Start Date/Time of Treatment*	*End Date/Time of Treatment*	*Study Day of Start of Treatment*	*Study Day of End of Treatment*
Char	*Char*	*Char*	*Num*	*Num*
50	*50*	*50*	*8*	*8*
TREATMENT	2012-12-01	2012-12-01	2	2
TREATMENT	2012-12-02	2012-12-02	0	3
TREATMENT	2012-12-03	2012-12-03	4	4
TREATMENT	2012-11-30	2012-11-30	1	1

- Negative test data contains issues

EPOCH	EXSTDTC	EXENDTC	EXSTDY	EXENDY
Epoch	*Start Date/Time of Treatment*	*End Date/Time of Treatment*	*Study Day of Start of Treatment*	*Study Day of End of Treatment*
Char	*Char*	*Char*	*Num*	*Num*
50	*50*	*50*	*8*	*8*
TREATMENT	2012-12-01	2012-12-01	2	-2
TREATMENT	2012-12-02	2012-12-02	0	-3
TREATMENT	2012-12-03	2012-12-03	-4	-4
TREATMENT	2012-11-30	2012-11-30	-1	1

Executability

- Fully executable
- Partially executable
  - FDAB057: When collecting ethnicity demographic data from clinical trial participants, the following two minimum choices should be offered: "HISPANIC OR LATINO" or "NOT HISPANIC OR LATINO"
- Not executable
  - FDAB055: Trial participants should self-report race and ethnicity and they should not be assigned by the study team.

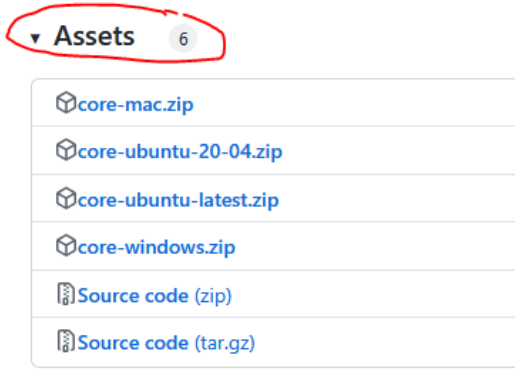


Technical Implementation of CORE

How can I run CORE or implement it in my own software?

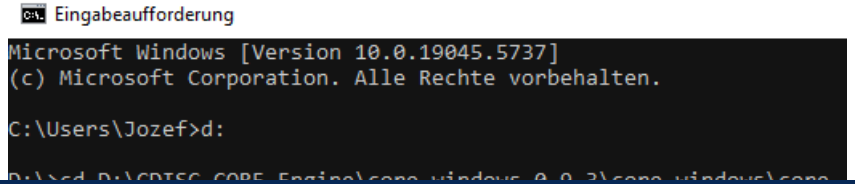
CORE executables

- All CORE code can be obtained from the CORE Github site
  - CORE is mostly written in Python
- Executables (Windows, Mac, Ubuntu-Linux) available for each published version
- Quick start through a "Command Line Interface" (CLI)
- Excellent documentation through "-help" commands
- Always choice between running as Python or using the executables



Using the Command Line Interface

- Use "cmd" or Powershell (Windows) to start the CLI
- Navigate to the folder where CORE was put



- When using Wir
- When using the

```
PS D:\PycharmProjects\cdisc-rules-engine> python core.py --help
Usage: core.py [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  list-ct                Command to list the ct packages available in the...
  list-dataset-metadata  Command that lists metadata of given datasets.
  list-rule-sets
  list-rules
  test
  update-cache
  validate               Validate data using CDISC Rules Engine
  version
```

Using the Command Line Interface: simple validation

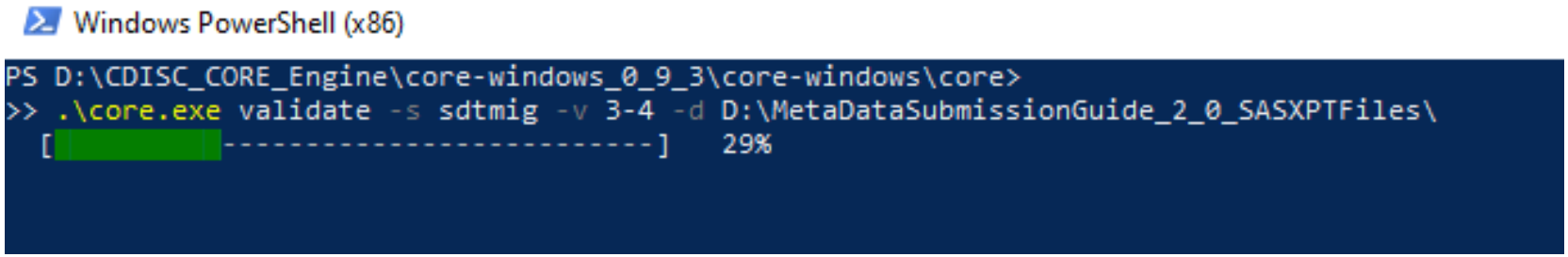
- To learn about the validation parameters, use:
  ".\core.exe validate --help"
  or
  "python core.py validate --help"

- The most important ones are:
  - -s: standard name, e.g. "sdtmig"
  - -v: standard version, e.g. "3-4"
  - -d: path where your XPT or Dataset-JSON files reside

Congatulations!
Your first CORE validation is running!

```
D:\CDISC_CORE_Engine\core-windows_0_9_3\core-windows\core>.\core.exe validate --help
Usage: core.exe validate [OPTIONS]

  Validate data using CDISC Rules Engine

  Example:

  python core.py -s SDTM -v 3.4 -d /path/to/datasets

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
  -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]
  -ss, --substandard TEXT         CDISC Substandard to validate against
  -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 [XLSX|JSON]
                                  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 [2-1|2-0|2.0|2.1]
                                  Define-XML version used for validation
  --whodrug TEXT                  Path to directory with WHODrug dictionary
                                  files
  --meddra TEXT                   Path to directory with MedDRA dictionary
                                  files
  --loinc TEXT                    Path to directory with LOINC dictionary
```

Your first CORE validation



This will generate a validation report in Excel format
in the same folder as where you have the software.
You can use "-of JSON" to get the report in JSON format
And "-o …" to get the report in another format

Many other options are available … Use them!

Getting the rules themselves

- You can get the rules themselves and storing them to file, using:

  ".\core.exe list-rules > rules.json"

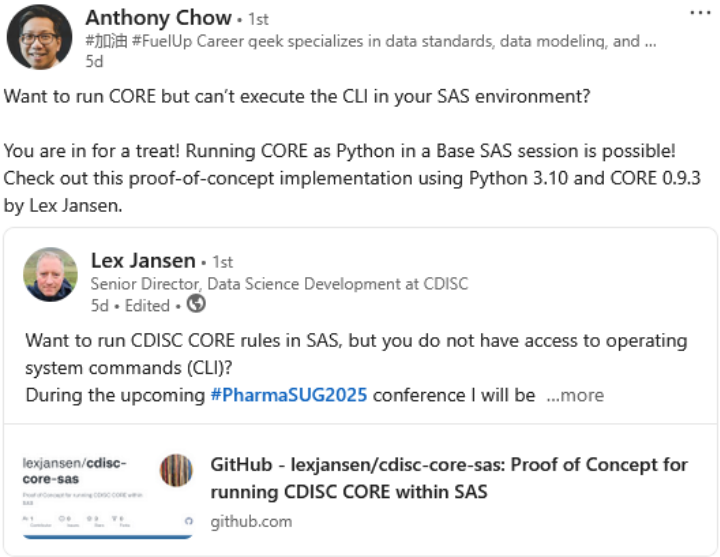
- This can be very interesting when developing a GUI on top of CORE

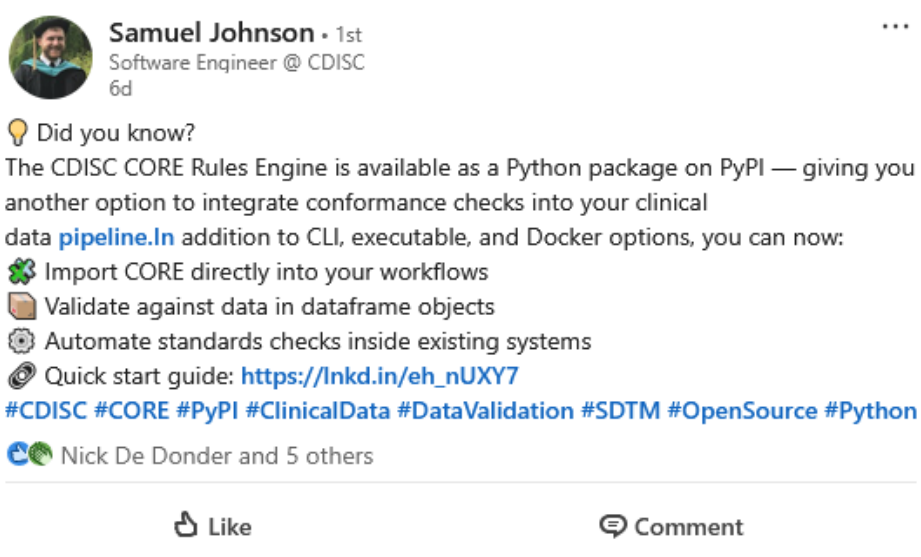
rules.json


```
[
    {
        "core_id": "CORE-000351",
        "author": "CDISC",
        "reference": [
            [
                {
                    "Citations": [
                        {
                            "Cited_Guidance": "Identifier used to uniquely identify a subject a
                            "Document": "IG v3.4",
                            "Item": "Specification",
                            "Section": "5.2"
                        }
                    ],
                    "Origin": "SDTM and SDTMIG Conformance Rules",
                    "Rule_Identifier": {
                        "Id": "CG0151",
                        "Version": "1"
                    },
                    "Version": "2.0"
                }
            ],
            [
                {
                    "Citations": [
                        {
                            "Cited_Guidance": "Identifier used to uniquely identify a subject a
                            "Document": "IG v3.2",
                            "Item": "Specification",
                            "Section": "5"
                        }
                    ],
                    "Origin": "SDTM and SDTMIG Conformance Rules",
                    "Rule_Identifier": {
                        "Id": "CG0151",
                        "Version": "1"
```

Implementing CORE in your own software

- Lots of background information on the LinkedIn "CDISC CORE Collaboration" forum https://www.linkedin.com/groups/12986635/





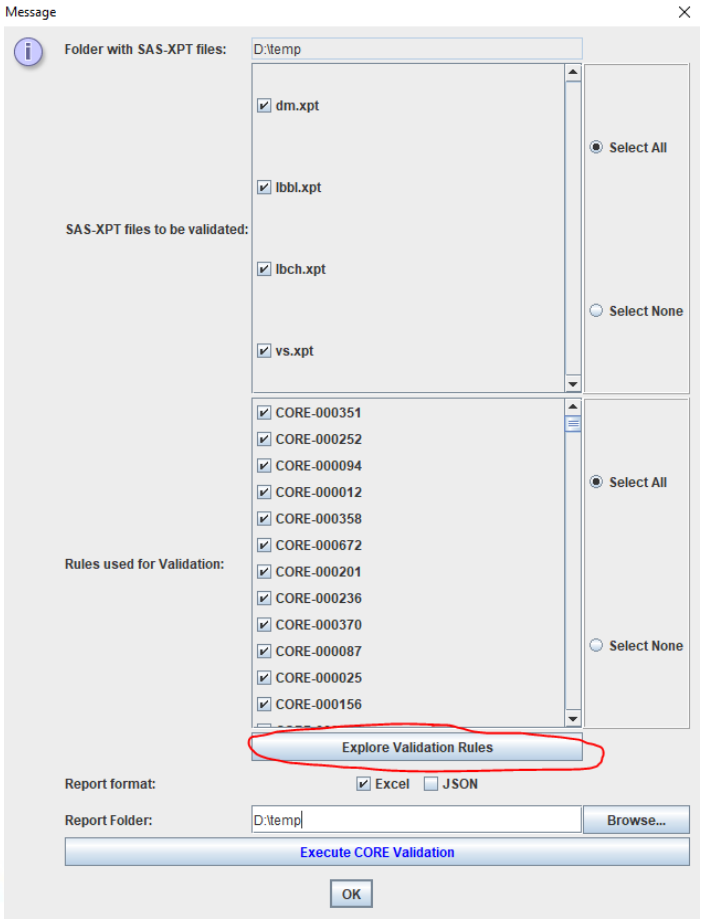
Implementing CORE in your own software
Many options ...

- Docker (see poster Sam Johnson)
- In existing Python applications (easy …)
- SAS: see Lex's post on LinkedIn
  and: https://github.com/lexjansen/cdisc-core-sas
- R: one can use the "reticulate" package
- Java / C# / …:
  Many software languages have a mechanism to start an external process in a separate thread.

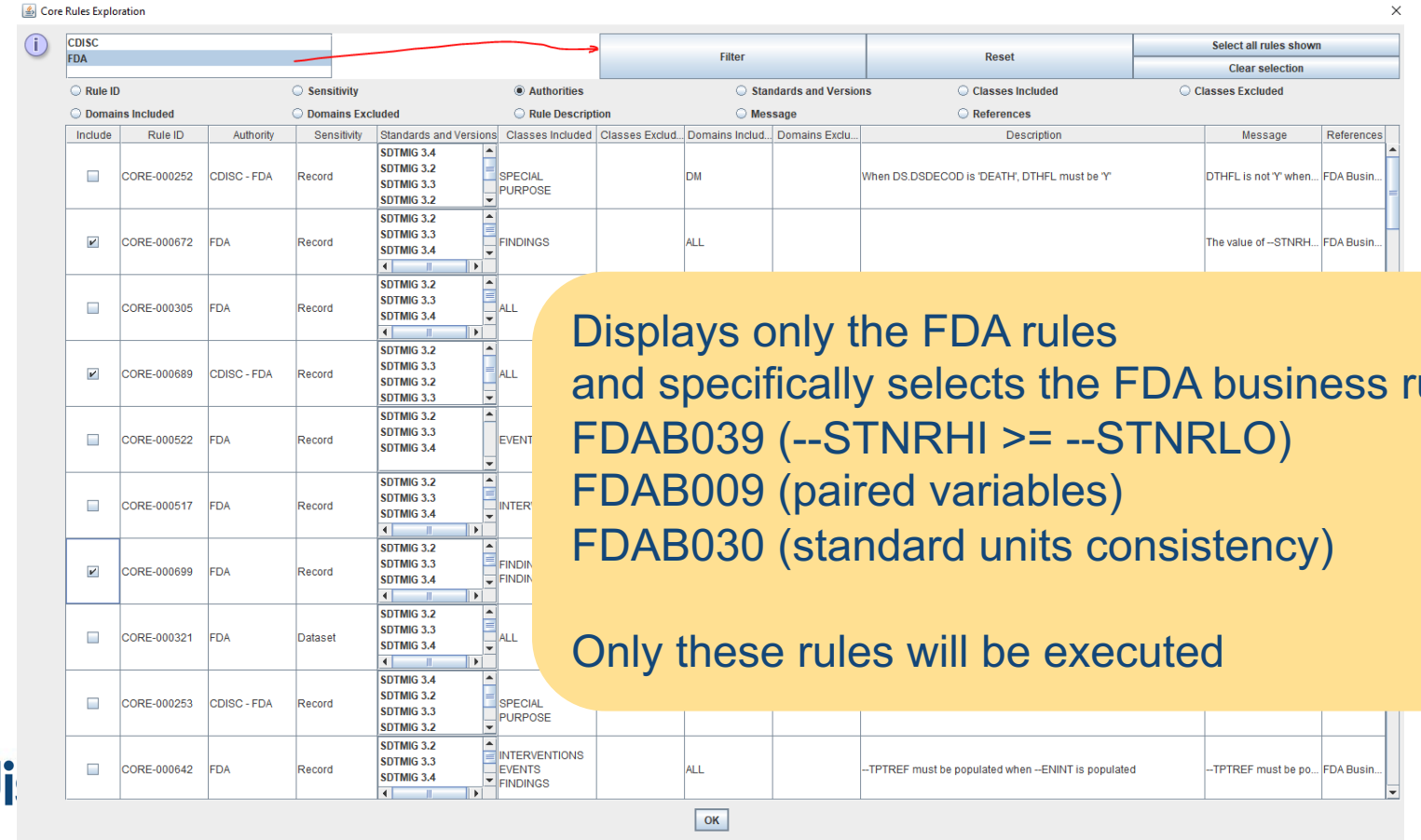
Implementing CORE in your own software - example

- One can use the generated "rules.json" file to allow selection of (groups) of rules to be applied on all, or a selection of, the XPT or Dataset-JSON files
- We have implemented this in several of our software offerings
- Essentially, the only limitation is … lack of imagination

Implementing CORE in your own software - example



Implementing CORE in your own software: Rules filtering

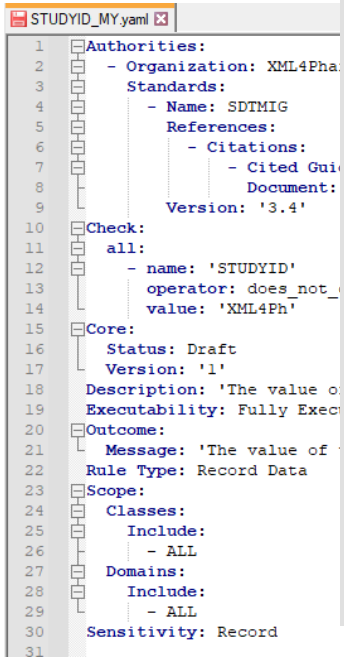


Displays only the FDA rules
and specifically selects the FDA business rules:
FDAB039 (--STNRHI >= --STNRLO)
FDAB009 (paired variables)
FDAB030 (standard units consistency)

Only these rules will be executed

Adding yo

- Many compa
- With CORE,




```yaml
Check:
  all:
    # throw an error when AGE (in years) is higher than 65
    - name: AGEU
      operator: equal_to
      value: YEARS
    - name: AGE
      operator: greater_than
      value: 65

Core:
  Id: XML4P1
  Status: Draft
  Version: '1'
Description: 'Age in years must be 65 or lower'
Executability: Fully Executable
Outcome:
  Message: 'Age in years must be 65 or lower'
Rule Type: Record Data
Scope:
  Classes:
    Include:
      - 'SPECIAL PURPOSE'
  Domains:
    Include:
      - DM
Sensitivity: Record
```

ılity assurance

YAML or JSON
er with your own

rameter to point
h your own rules

s sdtmig –v 3-4
s
XML4Pharma

Adding your own (QA) rules to CORE: Example Result

CORE-Report-2025-04-2...

	A	B	C	D	E	F	G	H	
1	CORE-ID	Message	Executability	Dataset	USUBJID	Record	Sequence	Variable(s)	Value(s)
2		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC001	1	1	STUDYID	CDISCPILOT01
3		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC001	2	2	STUDYID	CDISCPILOT01
4		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC002	3	1	STUDYID	CDISCPILOT01
5		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC002	4	2	STUDYID	CDISCPILOT01
6		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC002	5	3	STUDYID	CDISCPILOT01
7		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC002	6	4	STUDYID	CDISCPILOT01
8		The value of the STUDYID must contain or start with XML 4Ph	fully executable	ae.xpt	CDISC002	7	5	STUDYID	CDISCPILOT01

Conformance Details | Dataset Details | Issue Summary | **Issue Details** | Rules Report

CORE-Report-2025-04-2...

	A	B	C	D	E	F
1	CORE-ID	Version	CDISC RuleID	FDA RuleID	Message	Status
2		1			The value of the STUDYID must contain or start with XML 4Ph	SUCCESS
3						
4						
5						
6						
7						
8						

Conformance Details | Dataset Details | Issue Summary | Issue Details | **Rules Report**

One can also add the own rules to the "cache", i.e. add them to the CDISC/FDA rules

Automating the generation of custom rules

- Generation of custom rules can easily be automated, e.g. for:
  - CDISC Code Tables
  - Biomedical Concepts
  - LOINC to CDISC Mappings (>10,000 available)
- Examples:
  - When VSTESTCD=SYSBP,DIABP, VSORRSU must be one of mmHg, cmHg, Pa
  - When LBLOINC=1751-7, LBTESTCD must be "ALB", LBTEST must be "Albumin" LBSPEC must be "SERUM" or "PLASMA"

```
Check:
  all:
    - name: FATESTCD
      operator: equal_to
      value: 'ACMITYPE'

    - name: FASTRESC
      operator: is_not_contained_by
      value:
        - 'TYPE 1 MYOCARDIAL INFARCTION'
        - 'TYPE 2 MYOCARDIAL INFARCTION'
        - 'TYPE 3 MYOCARDIAL INFARCTION'
        - 'TYPE 4A MYOCARDIAL INFARCTION'
        - 'TYPE 4B MYOCARDIAL INFARCTION'
        - 'TYPE 4C MYOCARDIAL INFARCTION'
        - 'TYPE 5 MYOCARDIAL INFARCTION'
```

Thank You!

ndedonder@cdisc.org

jozef.aerts@xml4pharma.com