



The Clinical Data
Science Conference

EU 2024

10-13 November
Strasbourg Convention Centre



Implementing CDISC CORE in a cloud native, regulated environment supporting Rare Diseases submissions

- Emmy Pahmer, Senior Clinical Programmer, OCS Life Sciences
- Stijn Rogiers, Head Data Integration & Standards, argenx
- Sandeep Juneja, Clinical Solutions Lead, argenx



Authors



Stijn Rogiers

Head Data Integration & Standards, argenx

Stijn joined argenx in June 2022 as Head Data Integrations and Standards (Data Acquisition team). Argenx is a fast-moving and growing Biotech company. Stijn has 20+ years of experience both at CRO, Pharma industry and Technology. He worked 5 years at SGS Life Sciences (CRO), 10 years within Johnson & Johnson (Pharma) and 7 years at SAS Institute (analytics leader) before moving to argenx. Stijn is also a member of the European CDISC Coordinating Committee (E3C).



Emmy Pahmer

Senior Consultant, Clinical Programming, OCS Life Sciences

Emmy has been working in clinical research for over 20 years, primarily in SAS programming roles. She has experience with Data Management processes, PK/PD reconciliation and reporting, generating SDTM data as well as supporting other teams: Biostatistics, Medical Monitoring, Sample Management, Pharmacovigilance and Operations. Since August 2023, Emmy is supporting the Data Integration & Standards team at argenx.

Agenda

1. Introduction
2. Implementation Journey
3. Technical Implementation & Execution
4. Evaluation - CORE findings after running it on argenx ongoing studies
5. Next Steps



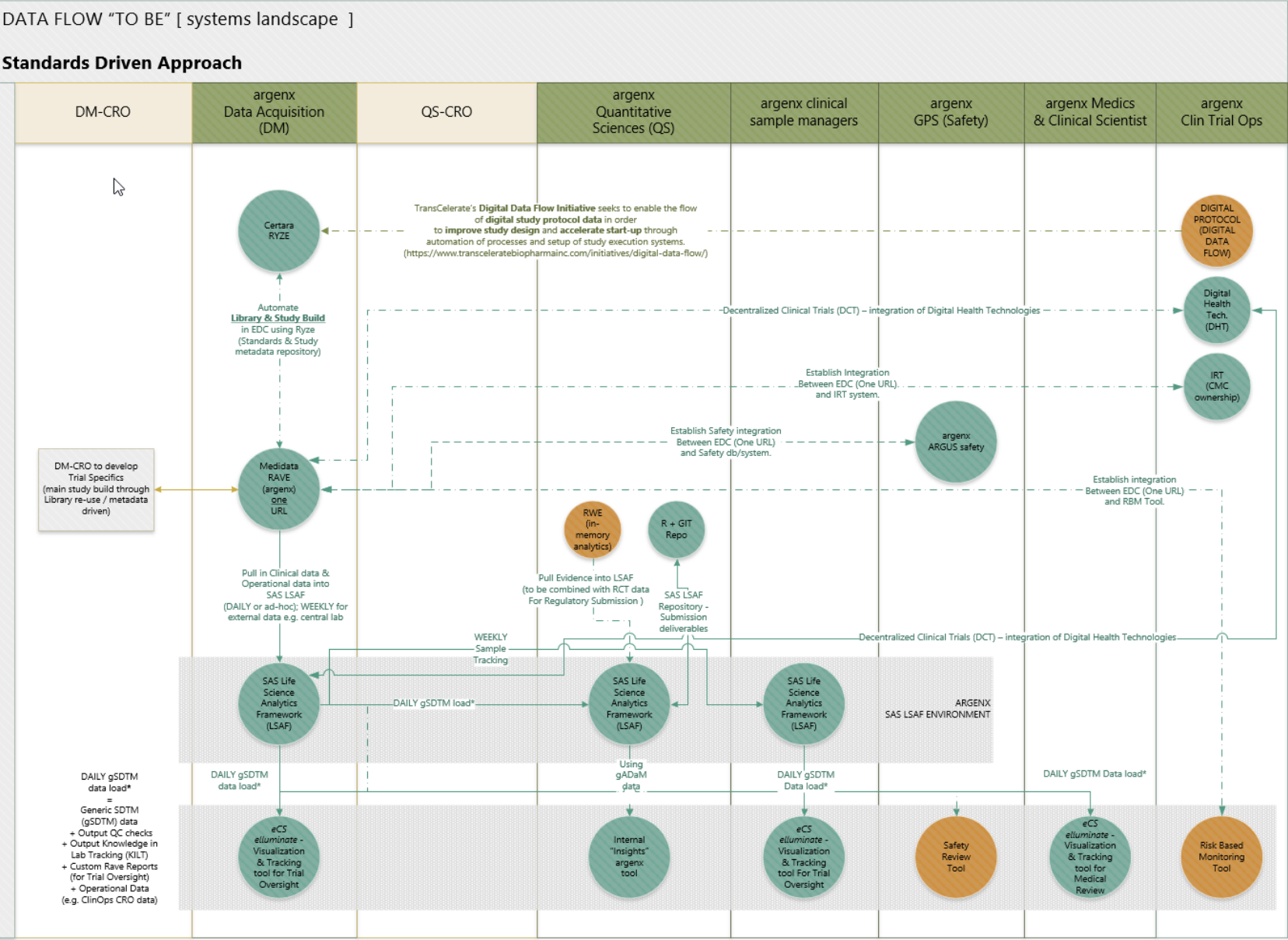
CDISC CORE Intro

The overall goal of the CORE Project is to deliver a **governed set of unambiguous and executable Conformance Rules** for each Foundational Standard, and to **provide a Reference Implementation** of an open-source execution engine for the executable Rules.

- **CDISC Open Rules** - over 1900 rules covering multiple versions of implementation guides for various Foundational Standards and Regulatory Rules
- **CORE Engine** – Open-source framework, available in github and free to all CDISC community. Release the open-source engine under the CDISC Open-Source Alliance (COSA)
- **CORE Rule Editor** – Web-based application to develop CORE rules in YAML with real-time syntax checking.

[CORE Github - https://github.com/cdisc-org/cdisc-rules-engine](https://github.com/cdisc-org/cdisc-rules-engine)

argenx Environment



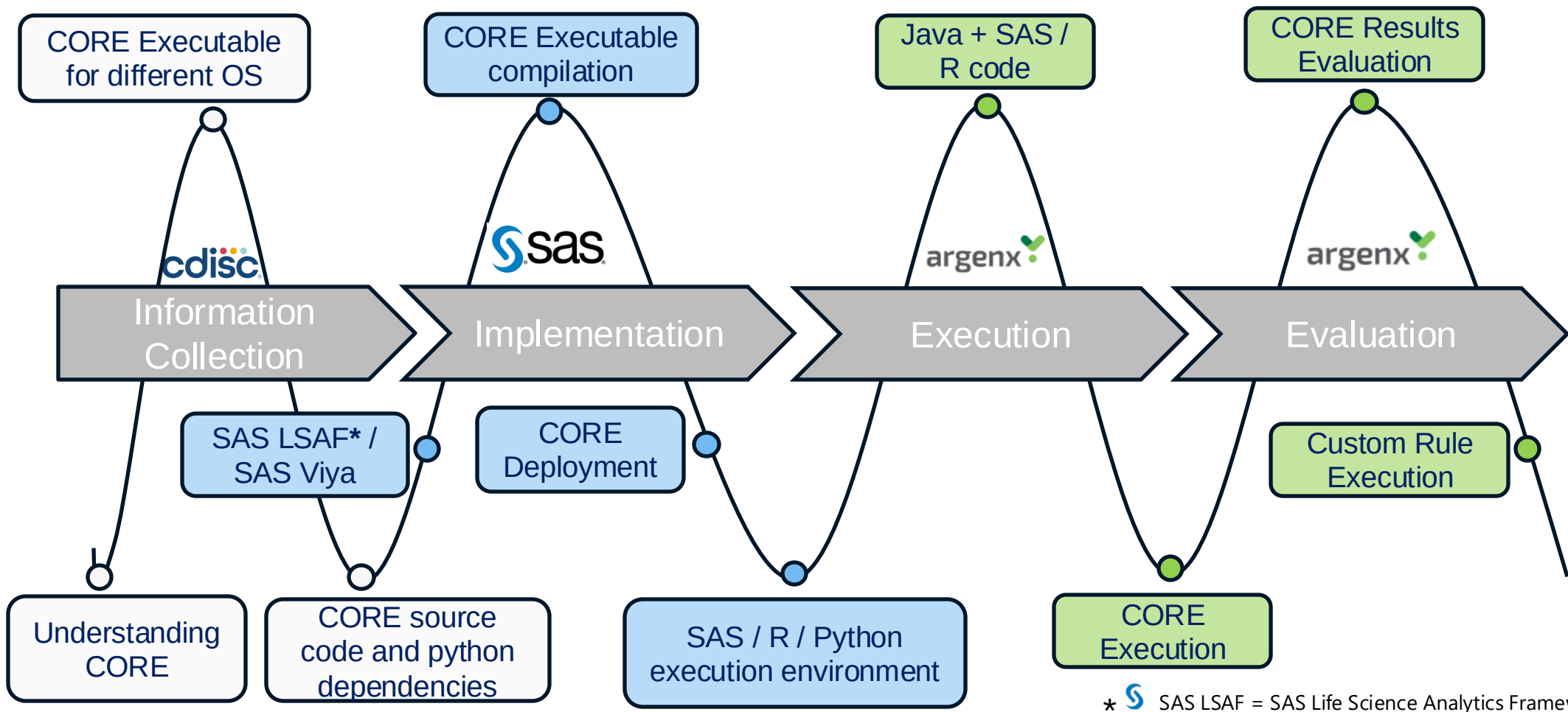
Why CORE?

- No data transfer in cloud
- Comprehensive & flexible
 - CDISC Open rules
 - Regulatory rules
 - Custom in-house rules
 - Execution flexibility on subset data
- Open-source and transparent
- CORE adoption
 - CORE execution by pharma / biotech in their environments
 - Research collaboration with regulatory agency for their business rules creation
- Free



Innovation: Our Core Mission
We live to innovate — and do so at every step.

Implementation Journey



*  SAS LSAF = SAS Life Science Analytics Framework

Execution – SAS / Java

- Java method
 - Pass parameter list ProcessBuilder object

```
public void Execute(String cdisc_core_root_dir, String core_cmd, String core_standard, String core_standard_version, String resource_template, String study_xpts, String result_file) {  
  
    List<String> params = Arrays.asList(cdisc_core_root_dir, core_cmd, "-s", core_standard, "-v", core_standard_version, "-rt", resource_template, "-d", study_xpts, "-o", result_file);  
    ProcessBuilder pb = new ProcessBuilder(params);  
    pb.redirectErrorStream(true);  
  
    try {  
        Process p = pb.start();  
        BufferedReader reader = new BufferedReader(new InputStreamReader(p.getInputStream()));  
        String s;  
        while ((s = reader.readLine()) != null) {  
            System.out.println(s);  
        }  
    } catch (IOException e) {e.printStackTrace();}  
}
```

- SAS macro
 - Call Java method using JavaObj

```
%macro run_cdisc_core_checks(core_cmd=,core_standard=,core_standard_version=, study_xpts=, study_report_path=, study_report_name=);  
%let resource_template=%str(&cdisc_core_root_dir./resources/templates/report-template.xlsx);  
%put resource_template = &resource_template;  
  
%if %length(&study_report_name.)<1 %then %let study_report_name = CDISC_Core_Results_%SYSFUNC(translate(%SYSFUNC(datetime()),E8601DT),'-','')).xlsx;  
%let result_file = %str(&study_report_path./&study_report_name.);  
%put result_file = &result_file;  
  
* Initialize the Classpath;  
%init_classpath_update;  
* Add the jar file to the Classpath;  
%add_to_classpath(&base_path./jar/run_cdisc_core.jar);  
  
data cdisc_core_cli;  
    dcl javaobj u("lsaf_cdisc_core/Run_CDISC_Core");  
    u.callVoidMethod("Execute3",kstrip("&cdisc_core_root_dir/core"),kstrip("&core_cmd"), kstrip("&core_standard"),kstrip("&core_standard_version"),  
                    kstrip("&resource_template"),kstrip("&study_xpts"), kstrip("&result_file"));  
run;  
*Reset to Original ClassPath;  
%reset_classpath;  
%mend run_cdisc_core_checks;  
%run_cdisc_core_checks(core_cmd=&core_cmd.,core_standard=&core_standard.,core_standard_version=&core_standard_version.,  
                        study_xpts=&study_xpts., study_report_path=&study_report_path., study_report_name=&study_report_name.);
```

- LSAF UI (Jobs)

Name: test_run_cdisc_core.job
Documentation:

Label	Value
CORE Command	validate
CORE Standard	sdrtmig
CORE Standard Version	3-4
Study Data (xpt files) location	☐ /general/share_general/utill
Study Report path	☐ /general/share_general/utill
Study Report Name	argx_test_study

Execution – R (within LSAF)

- R Code

```
1 # check if its transient path, switch to workspace path and if its workspace, use workspace path
2 ws_path <- ifelse(grepl(".transient",LSAF_WS), paste("/lsafshared/SASWorkspaces/", LSAF_USR, sep=""), LSAF_WS)
3
4 cdisc_core_utility <- ifelse(exists("cdisc_core_utility"), cdisc_core_utility, paste(ws_path, "/general/share_general/utilities/cdisc_core/executable", sep=""))
5 # cdisc_core_utility <- paste(LSAF_WS, "/Customer Preview/CDISC_Core/core", sep="")
6 # print(cdisc_core_utility)
7
8 core_script <- paste("",cdisc_core_utility, "/core", "", sep="")
9 # print(core_script)
10
11 core_cmd <- ifelse(exists("core_cmd"), core_cmd, "validate")
12
13 core_standard <- ifelse(exists("core_standard"), core_standard, "sdtmig")
14 core_std_cmd <- paste("-s", core_standard)
15
16 core_standard_version <- ifelse(exists("core_standard_version"), core_standard_version, "3-4")
17 core_stdv_cmd <- paste ("-v", core_standard_version)
18
19 core_report_template <- ifelse(exists("core_report_template"), core_report_template, paste(cdisc_core_utility, "/resources/templates/report-template.xlsx", sep=""))
20 core_rptt_cmd <- paste("-rt ", core_report_template, "", sep="")
21
22 study_xpts <- ifelse(exists("study_xpts"), study_xpts, paste(LSAF_WS, "/general/share_general/utilities/cdisc_core/source_code/test_data", sep=""))
23 study_data <- paste("-d '", study_xpts, "'", sep="")
24
25 study_report_path <- ifelse(exists("study_report_path"), study_report_path, paste(LSAF_WS, "/general/share_general/utilities/cdisc_core/source_code/reports", sep=""))
26 study_report_name <- ifelse(exists("study_report_name"), study_report_name, "test")
27 study_report <- paste("-o '", study_report_path, "/", study_report_name, "", sep="")
28
29 core_exec_cmd <- paste(core_script, core_cmd, core_std_cmd, core_stdv_cmd, core_rptt_cmd, study_data, study_report, sep=" ")
30 core_exec_cmd <- gsub("[\r\n]", "", core_exec_cmd)
31 print(core_exec_cmd)
32
33 system(core_exec_cmd)
34
35 rm(cdisc_core_utility, core_script, core_cmd, core_standard, core_standard_version, core_report_template, study_xpts, study_report_path, study_report_name)
36
```

- LSAF UI (Jobs)

Name: execute_core3.job

Documentation:

Label	Value
CORE Command	validate
CORE Standard	sdtmig
CORE Standard Version	3-4
Study Data (xpt files) location	☐ /general/share_general/utili
Study Report path	☐ /general/share_general/utili
Study Report Name	test

Implementation Challenges

- **Different Operating System**

- CDISC executable compiled for windows and Linux (ubuntu) whereas SAS LSAF uses CentOS Linux.
- CDISC source code needed to be compiled into executable for SAS LSAF CentOS by downloading all dependencies.

- **Execution Permission**

- CDISC CORE executable required execution permission which required working with SAS team.
- SAS LSAF does not allow command line execution using Base/SAS

- **Programming Language(s) support**

- Additional source code required to be written in Java, compiled into executable jar and wrapped into SAS macro to support CORE execution via SAS
- Source code written in R programming language too, to execute CORE in SAS LSAF.

- **Shared Workspace**

- Multiple users can access same executable

Report

- CORE executed and output reviewed for a number of trials in 2 steps, early in the year (v0.6.2) and recently (v0.8.1)
- New package of CDISC Open Rules was released between the 2 steps
- Only reviewed the rules where issues were reported

Conformance Details - Standards

Conformance Details	
Report Generation	2024-10-04T15:54:06
Total Runtime	11.25 seconds
CORE Engine Version	0.8.1
Standards Details	
Standard	SDTMIG
Version	V3.3
CT Version	
Define-XML Version	2.1.0
UNII Version	
Med-RT Version	
MedDRA Version	
WHODRUG Version	
SNOMED Version	

Dataset details – datasets used, size and number of records

Dataset	Label	Location	Modified Time Stamp	Size (kb)	Number of Records
ae.xpt	Adverse Events	..\data	2024-10-03T01:29:12	11.44	9
be.xpt	Biospecimen Events	..\data	2024-10-03T01:29:12	9.52	31
cm.xpt	Concomitant/Prior Medications	..\data	2024-10-03T01:29:12	67.76	142
co.xpt	Comments	..\data	2024-10-03T01:29:12	1.76	0
da.xpt	Drug Accountability	..\data	2024-10-03T01:29:12	19.2	96
dc.xpt	Demographics as Collected	..\data	2024-10-03T01:29:12	9.12	19
dm.xpt	Demographics	..\data	2024-10-03T01:29:12	8.88	18

Issue Summary – Shows the number of cases found for each issue. Explanation column to capture follow-up comments

Dataset	Dataset Domain	CORE-ID	Message	Issues	Explanation
ae.xpt	AE	CORE-000553	AEENDY is not properly calculated per study day algorithm	8	
ae.xpt	AE	CORE-000550	Variable is not an allowed variable for an Observation Class	1	
ae.xpt	AE	CORE-000552	AESTDY is not properly calculated per study day algorithm	9	
be.xpt	BE	CORE-000464	BEDY is not calculated correctly even though the date portion of BEDTC is complete, the date portion of DM.RFSTDTC is a complete date, and BEDY is not empty.	31	
be.xpt	BE	CORE-000550	Variable is not an allowed variable for an Observation Class	1	

Report

Issue Details – Detailed list of issues, for each case encountered

CORE-ID	Message	Executability	Dataset	USUBJID	Record	Seque	Variable(s)	Value(s)
CORE-000200	when PCSTAT is null or PCDRVFL not equal to 'Y'	fully executable	PC	0010013001	21	21	PCDRVFL, PCORRES, PCSTAT	Not in dataset, ,
CORE-000200	when PCSTAT is null or PCDRVFL not equal to 'Y'	fully executable	PC	0010013001	22	22	PCDRVFL, PCORRES, PCSTAT	Not in dataset, ,
CORE-000200	when PCSTAT is null or PCDRVFL not equal to 'Y'	fully executable	PC	0010013001	23	23	PCDRVFL, PCORRES, PCSTAT	Not in dataset, ,
CORE-000200	when PCSTAT is null or PCDRVFL not equal to 'Y'	fully executable	PC	0010013001	24	24	PCDRVFL, PCORRES, PCSTAT	Not in dataset, ,

Rules Report – Rules executed Successfully vs Skipped. Link between CORE-ID & CDISC Open Rule

CORE-ID	Version	CDISC RuleID	FDA RuleID	PMDA RuleID	Message	Status
CORE-000028	1	CG0008			--TPTREF is empty and --ELTM is not empty	SKIPPED
CORE-000237	1	CG0027			--SCAT is equal to --CAT	SUCCESS
CORE-000201	1	CG0029			USUBJID is not found in DM.USUBJID	SUCCESS
CORE-000168	1	CG0034			VISITNUM is not among VISITNUM in SV domain.	SUCCESS
CORE-000012	1	CG0040			AEOCCUR is present in AE dataset.	SUCCESS
CORE-000022	1	CG0041			At least one of the Seriousness criteria (AESCAN, AESCONG, AESDISAB, AESDTH, AESHOSP, AESLIFE, AESOD or AESMIE) = 'Y', but AESER = 'N' or empty.	SUCCESS

Evaluation - Challenges

Define.xml

- CDISC Open Rules are executed with or without Define.xml file

Conformance Details	
Report Generation	2024-10-22T10:20:23
Total Runtime	72.91 seconds
CORE Engine Version	0.8.1
Standards Details	
Standard	SDTMIG
Version	V3.3
CT Version	
Define-XML Version	

- If non-well-formed Define.xml was present, CORE did not execute the rules and the log provided a general message about the Define.xml

```
[ERROR 2024-10-08 11:55:15,840 - excel_report.py:110] - Unknown value DefineVersion in ValueSet. Unable to check value.
Traceback (most recent call last):
  File "cdisc_rules_engine\services\define_xml\base_define_xml_reader.py", line 80, in read
  File "odmlib\loader.py", line 28, in MetaDataVersion
  File "odmlib\define_loader.py", line 68, in load_metadataversion
  File "odmlib\define_loader.py", line 25, in load_document
  File "<string>", line 1, in <module>
  File "odmlib\odm_element.py", line 82, in __init__
  File "odmlib\odm_element.py", line 91, in __setattr__
  File "odmlib\typed.py", line 277, in __set__
  File "odmlib\valueset.py", line 78, in value_set
ValueError: Unknown value DefineVersion in ValueSet. Unable to check value.
```

Evaluation - Challenges

CORE Log

- Processing is not sequential (one check completing then the next starting) resulting in difficulty to read and process log information for checks

```
[INFO 2024-10-22 11:01:56,662 - console_logger.py:43] - is_suitable_for_validation. rule id=CORE-000002, domain=AE, result=False
[INFO 2024-10-22 11:01:56,662 - console_logger.py:43] - is_suitable_for_validation. rule id=CORE-000001, domain=SUPPPR, result=False
[INFO 2024-10-22 11:01:56,662 - console_logger.py:43] - Skipped domain AE.
[INFO 2024-10-22 11:01:56,662 - console_logger.py:43] - Skipped domain SUPPPR.
```

- No information in log on why rules are skipped

CORE-000001	1	CG0176		IECAT equals "INCLUSION" and IERRES is not equal to "N".	SKIPPED
CORE-000002	1	CG0208		SESTDTC is required.	SUCCESS

Evaluation - Challenges

Rules Logic evaluation

- A few checks didn't work as expected, e.g.

CORE-ID	Message	Dataset	USUBJID	Variable(s)	Value(s)
CORE-000281	CMSTDTC is after CMENDTDC	CM	ARGX-113-2308-0010006005	CMENDTDC, CMSTDTC	2023-08-15, 2023-08

Compared partial dates, included them when they were equal. **From initial review, corrected now.**

CORE-ID	Message	Dataset	USUBJID	Variable(s)	Value(s)
CORE-000200	RSORRES cannot be null when RSSTAT is null or RSDRVFL not equal to 'Y'	RS	ARGX-113-2308-0010006005	RSDRVFL, RSORRES, RSSTAT	Y,,

ORRES is null because the result is derived (DRVFL=Y). The check is not taking the derivation flag into consideration.

CORE-ID	Message	Dataset	USUBJID	Variable(s)	Value(s)
CORE-000464	QSDY is not calculated correctly even though the date portion of QSDTC is complete, the date portion of DM.RFSTDTC is a complete date, and QSDY is not empty.	qs.xpt	ARGX-113-2308-0010006005	\$val_dy, QSDTC, QSDY, RFSTDTC	-1.0, 2024-07-22T09:40, 1.0, 2024-07-22T12:24

Procedure and reference date are the same. QSDY was calculated as 1 in the data. The check indicates that it should be -1.

Evaluation - Challenges

Rules Logic evaluation

- Issues were discussed with the CORE team
- Useful for them to get feedback on running the checks with real data
- Corrections have been made based on feedback 
- There are some new issues in the latest rules

Next Steps



Co-Creation: We Create Through Collaboration
We trust in the power of the team and know
that together we are better.

Qualification of CDISC CORE

- LSAF is GxP qualified environment.
- To use CDISC CORE in production environment, it requires to pass through argenx qualification process
- Qualification of SAS / R utilities to execute CDISC CORE

Note: *Qualification of CORE rules engine is out of scope*

- Working with CDISC team on automating the CI/CD for SAS LSAF OS
- Involvement in development of CORE engine and CDISC Open Rules
- Working with SAS team to support CORE execution in SAS LSAF
- Collaboration with CRO partners

Community effort, industry needs to provide feedback to CORE



Thank You

Emmy Pahmer - epahmer@argenx.com / emmy.pahmer@ocs-consulting.com

Stijn Rogiers – srogiers@argenx.com

Sandeep Juneja – sjuneja@argenx.com