Paper CD11

Using CDISC Validation Tools in a Validated Hosted Environment

Sandeep Juneja, SAS Institute Inc, Cary, NC

ABSTRACT:
Regulatory submission agencies worldwide request that submission data be delivered in CDISC format. Validating submission data to confirm that it adheres to CDISC standards is a big challenge, but it is a vital step in the pre-submission process because regulatory agencies run compliance checks against the submission data that they review.

Validation tools like SAS Clinical Standards Toolkit (SAS/CST) and OpenCDISC Validator can be easily operated in open environments by running SAS Macros and by issuing operating system commands to invoke the Java executable JAR files, respectively. However, users of validated hosted environments – like SAS Drug Development – are not allowed to issue operating system commands or access the SAS installation location, and require another way to execute the Validation tools.

This paper describes the development of a utility that uses the SAS Java Object to invoke the OpenCDISC Validator in a SAS session within the validated hosted environment of SAS Drug Development. It talks about the SASV9.cfg file customizations that the utility requires, and details the parameters that are needed to invoke the Validator.

This paper also describes customization required in order to use SAS Clinical Standards (SAS/CST) toolkit from validated hosted environment like SAS Drug Development.

NOTE: for brevity, this paper outlines only some of the key steps in the process. A fuller coverage is available in the companion slides on the SAS Drug Development Forum: https://communities.sas.com/docs/DOC-7781

Validation Tools

Although there are various validation tools available, the most prominent are OpenCDISC Validator and SAS Clinical Standards Toolkit(SAS/CST), which are widely used for compliance checks and content validation.

SAS Clinical Standards Toolkit (SAS/CST)

The SAS® Clinical Standards Toolkit provides support of multiple CDISC standards. It is an open-source solution. Most of the code is either Base SAS or SAS macro code. For XML-based standards, some JAVA and XSLT code is used.

OpenCDISC

OpenCDISC is an open source community that focuses on creating frameworks and tools for the implementation and advancement of CDISC Standards. It has developed OpenCDISC Validator, an open-source tool that is freely available for compliance checks for CDISC models like SDTM and ADaM

Validated Hosted Environment:

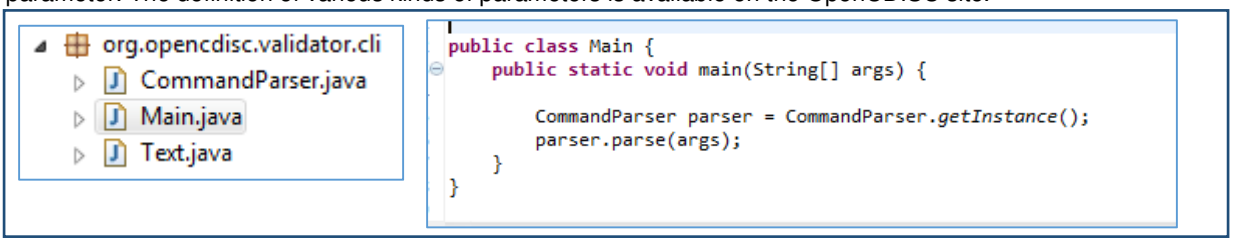
The FDA defines v*alidation* as, “Establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specification and quality attributes,” and *hosted environment* as, “A facility in which a third-party holds the data and runs the programs in its own computers.”
In order to make sure no unforeseen change happens to a hosted validated environment, systems are usually locked down and access is highly regulated.

SAS JavaObj:

The Java object provides a mechanism that is similar to the Java Native Interface (JNI) for instantiating Java classes and accessing fields and methods on the resultant objects. You can create hybrid applications that contain both Java and DATA step code.**OpenCDISC SAS Macro – Run_OpenCDISC.sas**

Java:

The main Java program that invokes the OpenCDISC Validator is Main.java, which is located in org.opencdisc.validator.cli package. Main.java contains a method named "main", which accepts a string array as a parameter. The definition of various kinds of parameters is available on the OpenCDISC site.



PC SAS:

Java Jar file can be referenced and executed from SAS using the following process.

SAS configuration file – sasv9.cfg

Specify the path of the Command Line OpenCDISC jar file - validator-cli-1.5.jar file into classpath using **JREOptions** parameter, as in this example.

`-JREOPTIONS (-Dsas.app.class.dirs=C:\SAS\test\OpenCDISC)*→ Path to validator-cli-1.5.jar file.` Specifying this option lets SAS access the jar file when SAS is invoked, and makes the jar file available throughout the SAS Session.

SAS Program

To call a java jar file from a SAS program, use the SAS JavaObj, as in this example.

```
data _opencdisc_cli;
dcljavaobj j ("org/opencdisc/validator/cli/Main");
array s{&i.} $200 (&cmd.);
j.callStaticVoidMethod("main",s);
run;
```

- JavaObj j references the Main class from the jar file.
- Array s takes in list of commands (list available on OpenCDISC website)
- Main method is invoked with Array s as parameter.

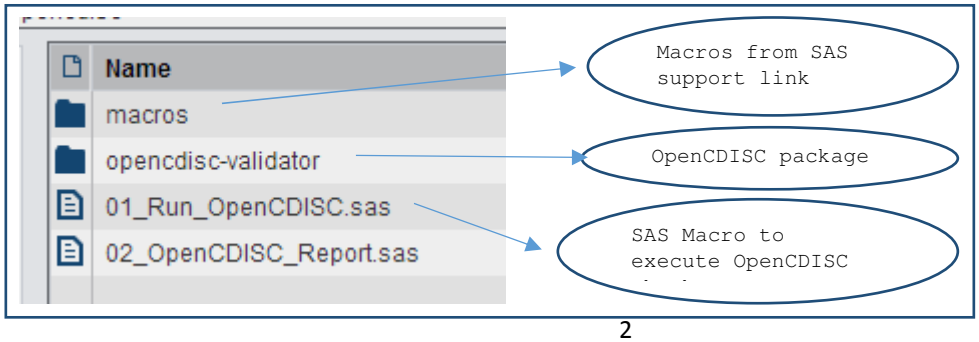
This code can also be developed as a SAS macro, and its path can be added to configuration file, as in this example.

`-insert sasautos "C:\SAS\test\OpenCDISC"*→ Path to SAS Macro calling OpenCDISC cli jar.`

The SAS Macro – Run_OpenCDISC.sas can be downloaded from the SAS Drug Development Forum link provided above

Hosted Environment:

In a validated hosted environment, the configuration file is read-only and is shared by all users. Any update to the configuration file could be prohibited. If this is the case, the jar file and path to the macro can be dynamically included in the SAS program itself without updating the SAS configuration file. The details for dynamically including the jar file are available in this link in the SAS Technical Support Web site: http://support.sas.com/kb/38/518.html Diagram provides snapshot of the setup of the macros in order to run OpenCDISC checks in hosted environment



SAS Clinical Standards Toolkit (SAS/CST)

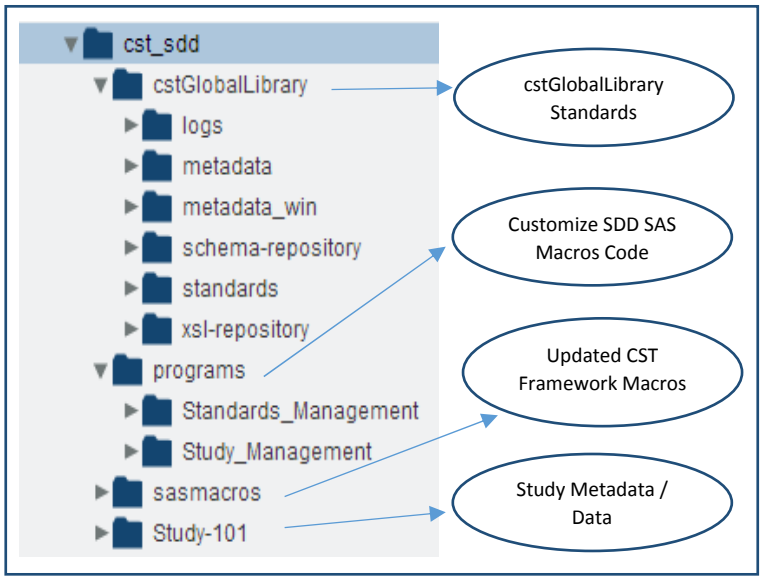
SAS/CST requires access to standards metadata (SAS datasets) and SAS macros in order to function. For a PC SAS installation, usually standards metadata is located on the PC drive, which can be easily updated, and SAS macros are located in SAS installation folder.

Hosted Environment:

Usually, in a hosted environment, both the standards metadata and SAS macros are located in restricted areas that end-users cannot modify. For such a case, registering any new custom standards or updating existing standards can be very challenging.

For a hosted system like SAS Drug Development (SDD), a repository location under Organization/Files (which is a global area where all users have at least read-only access) can be used to install standards metadata (for example, the cstGlobalLibraryfolder). The integrity of this location can be safeguarded by restricting update privileges to certain User Types or Group Members.

Below process describes on how SAS/CST can be setup in SAS Drug Development and can be used to register custom standards, update existing standards for custom domains, and carry out other toolkit related activities.



Global Standards:

Global standards metadata can be deployed under shared repository area as shown in figure.

CST Framework SAS Macros:

By default CST framework macros are located under !SASROOT/cstframework/sasmacros and CST is initialized by calling the %cstutil_setcstgroot macro, which contains a reference to the location of the cstGlobalLibrary global standards.

In order to make sure that the repository location of the global standards is referenced instead of default standards location, an update to the %cstutil_setcstgroot macro is required

```
%macrocstutil_setcstgroot(
    ) / des='CST: Set _cstGRoot macro variable';

%global _cstGRoot;

* Check Execution Environment - WINDOWS OR SAS DRUG DEVELOPMENT (SDD);
%if%symexist(_sasws_) %then%letenv=SDD; %else%letenv=WIN;
%putenv=&env.;

%if (&env.=WIN) %then%do;
%if&sysver=9.3%then %cstutilsetcstgroot93;
%else%let
_cstGRoot=%sysfunc(kcompress(%sysfunc(getoption(CSTGLOBALLIB)),%str(%")));
%end;
%else%do;
%let _cstGRoot=%str(&_sasws_/SAS/Files/cst_sdd/cstGlobalLibrary);
%end;
%mendcstutil_setcstgroot;
```

SDD Repository Location

By making change to just one macro, CST can be used to register customized standards, add customized domains to existing standards, generate study metadata, validate study data against any registered standard, generate define.XML, and even validate define.XML.

Validation Report – M&M Report

By default CST generates a .pdf report that contains only reference to the checkids. In order to clearly understand the actual test that the checkid refers to, another report is needed. By making a minor update to some of the CST framework SAS macros, such a report can be generated as an Excel Metrics and Matrix (M&M) spreadsheet.

There are 5 macros that generate various reporting metadata. By default these macros delete their work data sets when they finish executing. However, because these work data sets are needed to create the Excel M&M spreadsheet, we need to modify the modules so that they save a copy of them before their deletion.

```
data _cstGP;
        set &_cstResultsDSet;
run;

ods proclabel "General Process Reporting";
proc report data=&_cstResultsDset(where=(chec
        style(report)={just=center outputwidt
```

For example, the cstutil_reportgeneralprocess.sas module saves results in the _cstGP dataset. Similarly, other modules need to have a DATA Step added to save a copy of the work data set.

If all the metadata in the work data sets is available, then the Excel M&M spreadsheet can be generated using ODS Excel XP tagset.

Metrics Information:

Metrics details are similar to what is captured in the .pdf report. The information for this report comes from cstutil_reportprocessmetrics.sas macro.

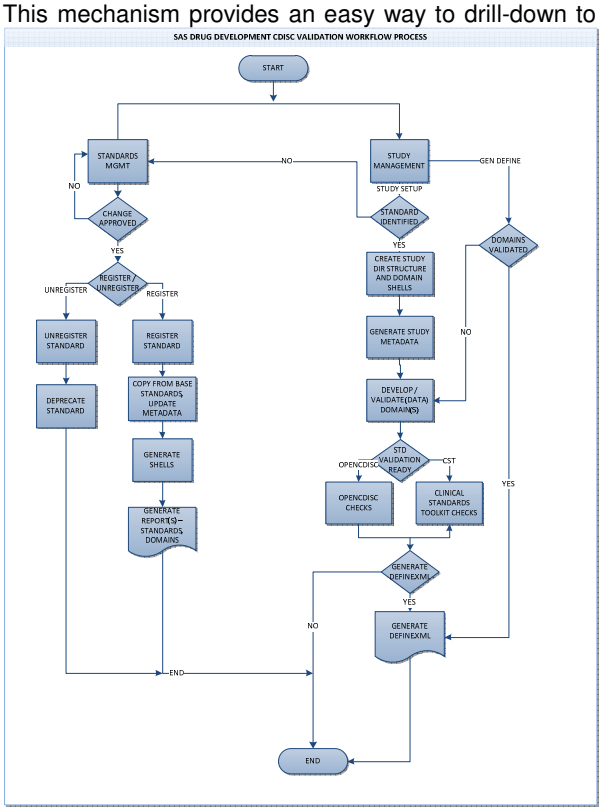
Summary Metrics		Table Metics				
Metric	#	Table	# Check Invocations	# Recs (if available	# Errors	# Check Invocations Not Run
# of distinct check invocations	229	AE	80	240	127	15
# check invocations not run	59	DM	81	260	95	15
Errors (severity=High) reported	13	EX	74	141	54	13
Warnings (severity=Medium) reported	473	SV	51	96	44	11
Notes (severity=Low) reported	599	TA	51	71	13	12
Structural errors, warnings and notes	4	TE	46	55	1	11
Content errors, warnings and notes	1130	TI	44	52	1	19
	.	TS	61	99	32	14
	.	TV	47	59	8	15
	.	VS	79	752	660	23
	.	ZZ	94	152	49	35

Matrix Information:

Matrix Information provides a snapshot about what checkids have been executed for which domains. It contains a description of the check, TableScope, ColumnScope, Severity, Execution Time required to execute the check and other details.

Check ID	Check Source	Message	TableScope	ColumnScope	Check Severity	Elapsed Time	CST	AE	DM	EX	SV	
SDTM0192	OpenCDISC	Observation date must be less than or equal to latest disposition date	EG+LB+VS	[**DTC][DSSTDTC]	Not Executed	0:00:00	⊗					
SDTM0193	OpenCDISC	Exposure end date must be less than or equal to latest disposition date	EX	[EXENDTC][DSSTDTC]	Not Executed	0:00:00	⊗					
SDTM0201	WebSDM	Null value in column	_ALL_	_ALL_	Pass	0:00:04				✓ (340/ 9)		✓
SDTM0201	WebSDM	Null value in column	_ALL_	_ALL_	Error	0:00:04		X (612/ 6)	X (280/ 10)		X (418/ 10)	
SDTM0202	SAS	Null value in column	_ALL_	_ALL_	Pass	0:00:04					✓ (418/ 10)	
SDTM0202	SAS	Null value in column	_ALL_	_ALL_	Note	0:00:04		⊟ (612/ 6)	⊟ (280/ 10)	⊟ (340/ 9)		⊟
SDTM0203	SAS	Character column value not upcased	ALL	ALL	Pass	0:00:03			✓ (280/ 10)	✓ (340/ 9)	✓ (418/ 10)	✓

Also, each check outcome under each domain is hyperlinked to a respective domain spreadsheet check location. This mechanism provides an easy way to drill-down to specific domain or specific check information and to correct the compliance failure that has been reported. Similarly, the M&M report can be produced from the OpenCDISC checks, which make it little easier to understand and debug the issues found.



The code for both generating M&M report for OpenCDISC and toolkit can be downloaded from below location https://communities.sas.com/community/support-communities/sas-drug-development

SAS Drug Development Workflow:

SAS Drug Development (SDD) also provides support for workflow implementation. Generic workflow diagram for executing both OpenCDISC and SAS/CST checks in SDD, it can be implemented as BPMN 2.0 workflow. It provides mechanism to encapsulate the complexities behind implementing the toolkit or OpenCDISC checks and can be presented to end user as a simple list of tasks. All the modification actions performed using this mechanism are audited and contains details about any changes made to the standards during its lifecycle. Furthermore it provides the functionality of versioning the standards there by provides the flexibility to revert to previous versions of standards if required.

CONCLUSION

OpenCDISC or SAS/CST checks can be executed even in a validated hosted environment in controlled process without having to download data to PC to validate data.

ACKNOWLEDGMENTS

I would like to thank Frank Roediger, who provided his expertise in making above described process work in SAS Drug Development.

REFERENCES

OpenCDISC http://www.opencdisc.org/download
Clinical Standard Toolkithttp://support.sas.com/documentation/cdl/en/clinstdtktug/66870/PDF/default/clinstdtktug.pdf
SAS Drug Developmenthttp://www.sas.com/en_us/industry/life-sciences/drug-development.html
SAS JavaObj: http://support.sas.com/rnd/base/datastep/dot/javaobj.html

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Sandeep Juneja
SAS Institute Inc
801 SAS Campus Drive
Cary, NC 27518
Phone:919-531-0541
Email: Sandeep.Juneja@sas.com
Web: https://communities.sas.com/community/support-communities/sas-drug-development