

Supporting CDISC Standards in Base SAS Using the SAS Clinical Standards Toolkit

Peter Villiers, SAS Institute Inc., Cary, NC

ABSTRACT

The use of regulatory standards aimed at clinical research data and metadata has become more commonplace over recent years. In the USA, the FDA is accepting standards from the Clinical Data Interchange Standards Consortium (CDISC) for submission of tabulation data (SDTM) and the main study metadata XML file (CRT-DDS). Work continues on new standards and updates to the existing ones. Previously, SAS users have not had to think so much about metadata and strict compliance to prescribed standards. The changing regulatory standards place an additional burden on SAS users, which translates to higher costs for companies.

This paper will explain how the SAS Clinical Standards Toolkit supports users by providing a framework of macro-based functionality to help ensure that standards are applied to clinical data and metadata. It will explain how users can create their own standards and use them within the framework. Finally, it will explain how updates to existing models will be supported as well as emerging standards will be incorporated over time.

INTRODUCTION

Over recent years, the frequency of submissions to regulatory agencies using CDISC standards has increased.

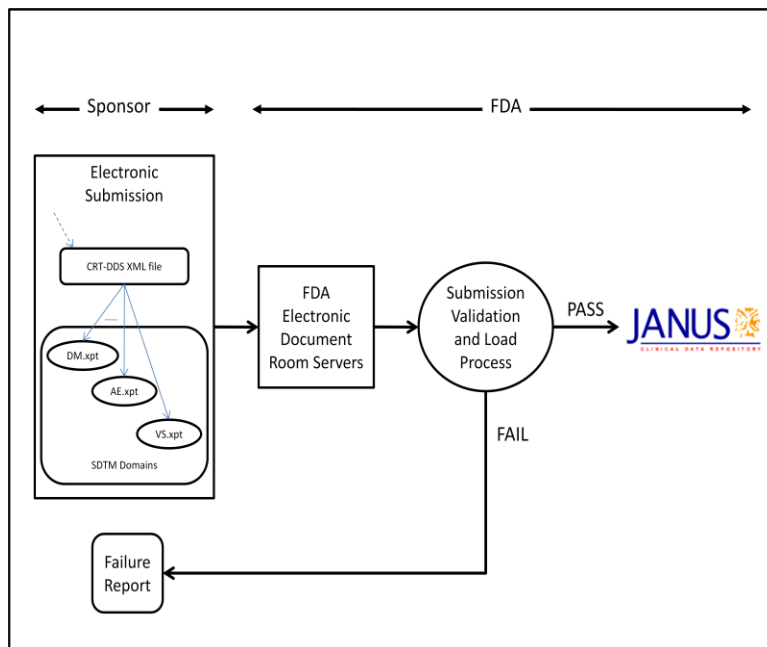


Figure 1 Current Data Flow for Electronic Submissions

Figure 1 shows the general flow of information when an electronic submission is made to the FDA. The electronic submission, based on the electronic common technical document (eCTD), contains pointers to study metadata files. The files, which are in the CDISC CRT-DDS XML file format, contain general information about the study along with metadata about the study data and pointers to the data files themselves. The data that supports tabulation reports can be submitted in the CDISC SDTM format and provided as SAS V5 transport files. At the time of writing, SDTM V3.1.1 is currently the accepted version for production submissions, and SDTM V3.1.2 is in the final stages of the CDISC review process, which involves the FDA.

The electronic submission is received by the FDA and stored on the electronic document room servers. Receipt and storage do not mean that the study is accepted.

Before the information is loaded into the JANUS warehouse, the FDA will subject it to a validation process and might reject the submission if all the required checks are not passed. After the data and metadata are loaded into JANUS, they are available for FDA reviewers to access using internal tools. Any delay in the acceptance of an electronic submission must be minimized. Therefore, it is important to expend effort up front to make sure that there is little chance of the data being rejected for technical reasons.

An important thing to note is that while SDTM describes a submission format for the data, there is a degree of variability allowed in the number and structure of the domains. This latitude permits data from different therapeutic

areas or study-specific data to be submitted and still be compliant with the overall SDTM version. This variability is commonly referred to as a study being *SDTM +/-*.

SAS CLINICAL STANDARDS TOOLKIT

The business goal of the SAS Clinical Standards Toolkit is to support the validation of standards used in regulatory submissions and, where needed, to convert metadata and data to and from the format required by the agencies. It must be flexible enough to handle the following:

- variations within a given standard and version, as with SDTM V3.1.1 +/-
- variations across release of a given standard
- new standards as they are developed and become required

A technical goal is to provide the functionality in a way that is familiar to SAS programmers so the learning curve can be lessened. To achieve this goal, users interact with the functionality via a series of macro calls and data sets.

To accommodate the business flexibility goal, the SAS Clinical Standards Toolkit has adopted a modular approach. A framework module handles the registration of standard-versions, like SDTM V3.1.1, and provides basic functionality that is used across all standard-versions. Modules that implement a standard version exist separately and can be installed; they can register themselves to the framework at any point. This approach has the benefit of allowing a new standard-version to be installed into a production system with no changes to existing standard versions. Additionally, the only changes to the underlying framework are the data sets that store the location of registered modules. It is expected that such changes will shorten the time required to validate the installation of a new standard version dramatically.

The SAS Clinical Standards Toolkit provides a consistent approach to validation across standards as well as standard versions. This paper will describe this approach within the context of the CDISC SDTM.

CDISC SDTM V3.1.1 VALIDATION

As mentioned above, the tabulation data sets that are submitted to the FDA must be compliant with the CDISC Submission Data Tabulation Model (SDTM). At a minimum, compliance means that companies will need to validate the SDTM data sets before inclusion in a submission. However, checking for compliance for internal purposes would also lead to consistent data that could be more easily aggregated for cross-study analysis.

Figure 2 shows the main inputs, controls, and outputs that are required when validating SDTM-based study data.

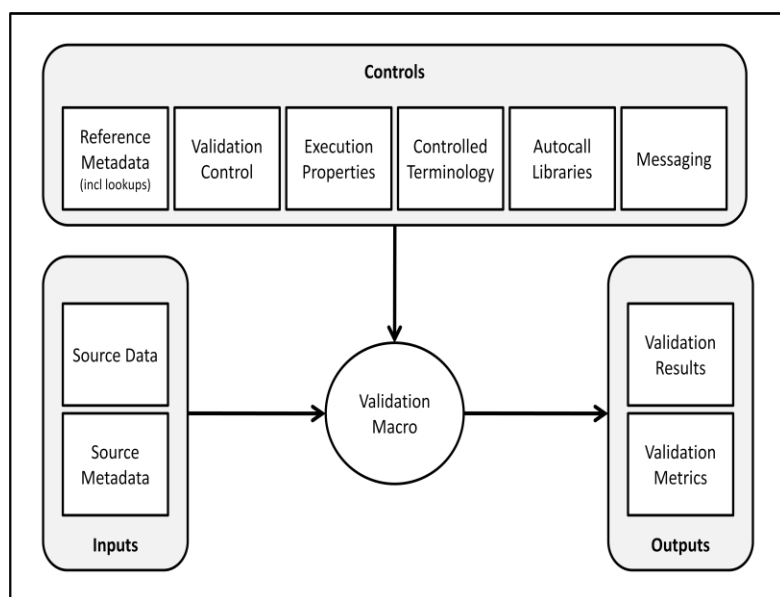


Figure 2 SDTM Validation

At first sight, this seems like a significant number of pieces; however, each has a unique job and many can be re-used across studies. Additionally, the SAS Clinical Standards Toolkit provides mechanisms that limit the need to specify these on each execution.

The **source [study] data** are the actual domains that need to be validated. The **source [study] metadata** are information about the structure of the domain data sets and the columns that each data set contains. This includes traditional PROC CONTENTS type metadata and also specific standard-version metadata.

To validate a study against a standard structure, metadata that describes the standard must exist. The specified metadata is called the **reference metadata**; it contains the same type of information as the source metadata with one addition. The reference metadata

also contains information about the allowed values for column contents.

The validation checks that are to be run are specified in a **validation control** data set. A registered standard that supports validation will provide the SAS implementation of the known set of validation checks. Not every possible check needs to be run at every point in time. Additionally, customer-level, study-specific checks might need to be executed for a run.

The CDISC organization provides controlled terminology that describes the data elements, the allowed values (submission values), and a preferred term. This standard is provided as a separate **controlled terminology** module that contains the raw data set as well as a SAS format library. This information is used in the validation process. Additionally, the study itself might have custom-controlled terminology that needs to be used in the validation process.

As the validation process runs, it creates a **validation results** data set that contains the outcome of all the validation checks that were executed. This data set can be used on its own to determine the success or failure of the validation and control processing accordingly. The SAS Clinical Standards Toolkit will also provide a number of prepared routines that can be used to generate reports off the results data set. The validation process can also create a **validation metrics** data set that contains identifying information and useful denominators such as the number of records per domain; subjects per domain; number of errors; and so on.

Along with the items described above, the SAS Clinical Standards Toolkit also uses the following:

- Message data sets. Standard versions supplied by SAS include these data sets. Customers can create and include their own data sets as the validation process runs.
- Property files. The SAS Clinical Standards Toolkit requires certain SAS global macro variables to execute. This mechanism also enables users to modify the execution parameters at runtime. Properties files allow modifications to be made in a more efficient manner than setting the initial values in a program.
- Autocall libraries. Each standard version that is supplied by SAS has an autocall library that implements macros in a manner specific to that standard. Users can specify additional autocall libraries to be included in the path at runtime.

This is a lot of information to provide to the validation process. The SAS Clinical Standards Toolkit uses a data set referred to as the **SAS References** data set to store these *pointers*. Among other items, it contains information about the standard; standard version; type and subtype of information being provided; and the location of the data sets and files.

EXAMPLE OF SDTM VALIDATION

The following is an example of an end-user program to validate SDTM domains.

```
/*
1. This sets up the root path for the study.
*/
%let studyRootPath=!sasroot\cstkit\sample\cdisc-sdtm-3.1.1\SASDemo;

/*
2. Set properties supplied as part of the CST-FRAMEWORK standard.
*/
%cst_setStandardProperties(
  _cstStandard=CST-FRAMEWORK
  ,_cstStandardVersion=1.2
  ,_cstSubType=initialize
);

/*
3. Load the initialization properties for this study.
This is primarily the location of the SASReferences file.
*/
%cst_setProperties(
  _cstPropertiesLocation=&studyRootPath.\programs\init.properties,
  _cstLocationType=PATH
);

/*
4. Allocate all the SAS references specified in the SASReferences file
```

```

*/
%cstutil_allocatesasreferences;

/*
5. Run the validation checks.
*/
%sdtm_validate;

```

It is immediately noticeable that the program itself is very short. The following paragraphs explain what is happening in the program and show the supporting files.

Line “1” merely sets a macro variable that defines the root path of the study. This setting is not necessary, but it is a convenience so that the study structure can be copied and there is only one variable to change.

Line “2” calls %cst_setStandardProperties to ensure that the required global macro variables are in place so that the framework can execute correctly. Each standard version that is supplied by SAS will have an initialization properties file to ensure that all the macro variables exist and are initialized for that standard version. In the case of the framework, the current initialization.properties file contains the following:

```

_cstDebug=0
_cstDebugOptions=mprint mlogic symbolgen mAutoLocDisplay
_cst_rc=0
_cst_MsgID=
_cst_MsgParm1=
_cst_MsgParm2=
_cstResultSeq=0
_cstSeqCnt=0
_cstResultsDS=work._cstResults
_cstSrcData=
_cstResultFlag=0
_cstFMTLibraries=WORK
_cstMessageOrder=APPEND
_cstCSTVersion=1.2
_cstMessages=work._cstMessages

```

A few of the more immediately useful macro variables are as follows:

- `_cstDebug`: allows the user to turn on debugging messages
- `_cstDebugOptions`: specifies which options to set
- `_cst_rc`: a return code set by every SAS Clinical Standards Toolkit macro
- `_cstResultsDS`: the default results data set to create

Line “3” sets the study-specific properties that will set any new or override any previously defined macro variables. The file contains the following:

```

_cstSASRefsLoc=&studyRootPath\control
_cstSASRefsName=sasreferences
_cstSASRefs=work._cstsasrefs

```

The first property sets the location (directory) for a persisted SASReferences file to be used. The second property sets the member name of the SASReferences file to be used because there might be more than one depending upon what the user is trying to achieve. The third property is the name of the working SASReferences file to be used - the concept of the working data set will be covered later.

Table 1, included at the end of the paper, shows the contents of a persisted SASReferences file, sorted by Type and SubType. This is how the validation process determines which information is used for a purpose.

The types and subtypes should be recognizable because they are similar to those mentioned in the preceding section. Columns in the table are as follows:

- `Standard` and `standardVersion`: These versions allow the SAS Clinical Standards Toolkit to look in the

standard version's registered metadata to find and *fill-in* the missing values for the standard's paths.

- Type and subType: This column indicates the reference type. This specification decouples the library and file naming that might be standardized at a site, from how it should be used in the SAS Clinical Standards Toolkit.
- SASRef and refType: These column names indicate whether this is a libref/fileref and the name that should be used.
- Path: The column identifies the location of the file set or data set to be used (if the LIBNAME or filename is not already available).
- Order: The order allows for precedence of multi-locational LIBNAMES, autocall libraries, and messaging data sets.
- MemName: This column identifies the name of the actual file or data set.
- Comment: for end-user benefit.

Interesting points that should be noted about this table include the following:

- A lot of the paths and memNames are empty. The toolkit has the ability to automatically look up these values from the standard version's registered metadata during the allocation process. (See the paragraph relating to Line "4" of the end-user program.)
- The study-specific items are placed higher in the ordering to be considered first in the process.
- External dictionaries, like MedDRA, can be used in the process and might be required for some validation checks to run.

The row with type and subType of control/validation refers to the list of checks that should be run. In this case, there is a persisted set of study-specific checks that are defined. This set contains a subset of the checks that are found in the registered SDTM-3.1.1 validation master data set. Table 2 shows a subset of the columns for this table, some of which are described here.

- CheckId: a unique identifier assigned to each check that is also used in reporting. A specific check SourceId might be defined in more than one check record. This allows greater granularity in reporting.
- Standard/StandardVersion: the StandardVersion is "****" because the check could be applied to any version of SDTM. If it were a specific version, only then would this have a version number.
- CheckSource/SourceId: Identify the origin of the check.
- CheckType: identifies if the check is run on a single value, across records within a table, or across tables.
- CodeSource: the macro that implements the check and will be executed to perform the checking. The values in the current row of the validation control data set are passed to the macro as a means to control the execution of the macro at that point in time. This, in combination with the autocall path in the SASReferences data set, allows users to write, include, and execute study-specific checks in exactly the same way as are checks supplied by SAS. This feature is powerful for user-customization.
- UseSourceMetadata: determines whether the sourceMetadata provided should be used during the checking or whether the code should use the reference metadata.
- TableScope: Identifies which domains are to be used for this check. Check 202 uses all domains, check 209 uses all "Findings" domains (this is determined from the source metadata), check 213 uses only the DM domain, and check 228 uses all domains that begin with "SUPP".
- ColumnScope: identifies which columns are to be used in the check. Check 202 uses any column sequence column, and check 209 specifies 2 columns.
- ReportAll: If set to "Y", the feature will report every occurrence in the data that fails the check. If set to "N", it will report only the first occurrence of the error and will also report that further occurrences will not be reported. This is useful where many of the same types of errors might be reported (which would *clutter up* the results data).

So consider check 202 in its entirety. It is based on a Janus check IR4004, which assures uniqueness of sequence numbers in the domains. It will generate an error record for every row in the domains that fail the check.

While this file might seem complex at first, it should be remembered that SAS will provide a significant number of pre-defined checks with the standard version based on the available regulatory documentation and additional analyses of the domains to be checked. Also, this effort to set up the checks that need to run happens very rarely and the validation control data set might even be shared across studies.

Returning to the end-user program, line "4" asks the framework to allocate all the filenames and libraries in the SASReferences file as well as set the autocall path and format search path accordingly. These actions include inserting the path and memName information from the registered standards to create the working SASReferences

file.

At this point, the environment is set up, the SASReferences data set has been processed, and all that is left to do is perform the validation. In line “5” the user performs the SDTM validation. The SASReferences file specified locations for the validation results and validation metrics.

Table 3 shows a subset of the validation results. Most of the domains passed without error; however, a couple of errors and warnings were observed. An end-user program can use this information to report on and if needed, make decisions on how to continue processing.

Table 4 shows an example of the validation metrics data set.

SUPPORT OF NEW VERSIONS, STANDARDS, AND USER-DEFINED STANDARDS

As versions of the CDISC SDTM standard are introduced, SAS Institute will release new modules to register to the SAS Clinical Standards Toolkit’s framework. The modules will be SAS implementations of the version. However, standards like SDTM have a degree of variability built into them to allow for customer, therapeutic area, or even study-specific modifications.

It is possible for a customer to define the reference metadata, custom terminology, custom checks, and associated messaging and then register them as a new standard version to the framework. The framework will treat these custom standards exactly as it treats those supplied by SAS Institute.

Given that a design philosophy was to apply a consistent approach to cross-cutting concerns like validation, it could be expected that a CDISC Analysis Data Model (ADaM) implementation would work in exactly the same way as SDTM.

CDISC CRT-DDS V1.0 (DEFINE.XML) CREATION

Currently, studies submitted electronically to the FDA must be accompanied by an XML file that conforms to the CDISC CRT-DDS standard. This file is more commonly called *define.xml* because it contains all the information that is found in the older define.pdf file, but is in a machine-readable format.

Constructing this file structure can be a time-consuming and error-prone exercise. The SAS Clinical Standards Toolkit uses the same approach for this as for the validation of standards. Figure 3 illustrates this approach.

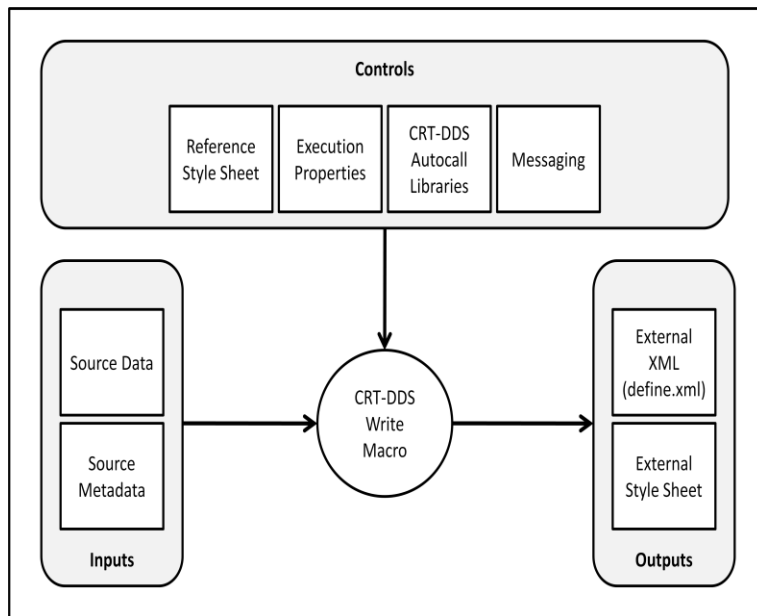


Figure 3 CRT-DDS Write

The **source data** is a tabular, data-set representation of all the information that will go into a CRT-DDS xml file. The **source metadata** describes the structure of the source data. SAS will provide a (default) **reference style sheet** that can be used to display the XML in a human readable format; however, a different style sheet can be specified by the user. The output of the write process is the **external XML** file [define.xml] and a copy of the reference style sheet referred to as the **external style sheet**.

The same SASReference file structure is used to provide this information to the CRT-DDS write macro. This also enables the user to specify the message data set to use, properties to set, and autocall path to use. It is important to understand that as more versions of CRT-DDS are supported, setting the autocall path to different versions will cause the updated XML structure to be produced.

EXAMPLE OF CRT-DDS CREATION CODE

```
/*
1. This sets up the root path for the study.
*/
%let studyRootPath=!sasroot\cstkit\sample\cdisc-crtdds-1.0\SASDemo;

/*
2. Set properties supplied as part of the CST-FRAMEWORK standard.
*/
%cst_setStandardProperties(
  _cstStandard=CST-FRAMEWORK
  ,_cstStandardVersion=1.2
  ,_cstSubType=initialize
);

/*
3. Load the initialization properties for this study.
This is primarily the location of the SASReferences file.
*/
%cst_setProperties(
  _cstPropertiesLocation=&studyRootPath.\programs\crtdds.properties,
  _cstLocationType=PATH
);

/*
4. Allocate all the SAS references specified in the SASReferences file
*/
%cstutil_allocatesasreferences;

/*
5. Run the validation checks.
*/
%crtdds_write;
```

The program is identical in structure to the one used to validate SDTM data. Step 3 uses a different properties file—this would allow a different SASReferences file to be used although this does not necessarily need to be the case. An example of this is shown in Table 5. Line “4” sets up the environment that needs to be in place. Line “5” calls the macro that takes the source data and metadata then uses them to create the CDISC CRT-DDS file, along with a style sheet in the desired location.

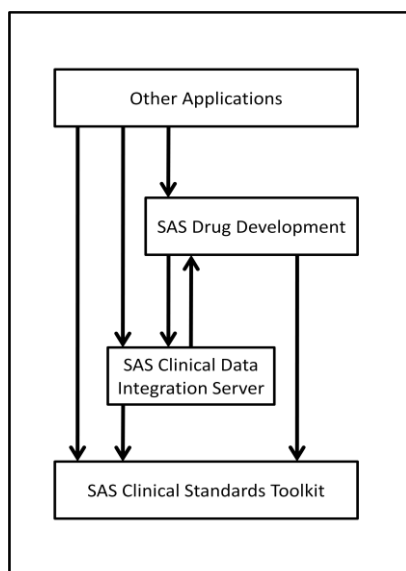


Figure 4 Example SAS Application Stack

Finally, it should be noted that because the same approach is taken to specify the (source and reference) data and metadata for CRT-DDS as with SDTM, it is possible to use the same process to validate the CRT-DDS data before it is used to create a define.xml file.

INTERACTION WITH OTHER SAS PRODUCTS

The functionality provided by the SAS Clinical Standards Toolkit is surfaced as macros that can be called from a SAS program. This means that any product capable of generating SAS code can call these macros. The SAS Clinical Standards Toolkit should therefore be considered a foundational piece within the SAS approach to Health and Life Sciences.

Figure 4 shows an example of how the SAS Clinical Standards Toolkit can be used within a larger application stack. The SAS Clinical Data Integration Server provides functionality to manage clinical standards metadata, study metadata, and the data integration capabilities for processing and preparing this data. It has the ability to consume the standards that are registered to the SAS Clinical Standards Toolkit. As a code generator, it can also generate the programs required to delegate the tasks to the SAS Clinical Standards Toolkit and execute them.

SAS Drug Development is a Web-based application that allows collaborative work among disparate teams. Part of its functionality is to provide a code execution environment that complies with 21CFR, pt.11 regulations. Programs can either be written completely within the web application, written manually and uploaded, or written in another tool and transferred to the application. If the SAS processing servers that support SAS Drug Development have had the SAS Clinical Standards Toolkit installed on them, then they will be able to take advantage of that functionality.

Finally, other SAS applications, or even customer applications, can take advantage of the SAS Clinical Standards Toolkit functionality by simply providing the required inputs and controls and then calling the macro API.

FUTURE STANDARDS DIRECTIONS

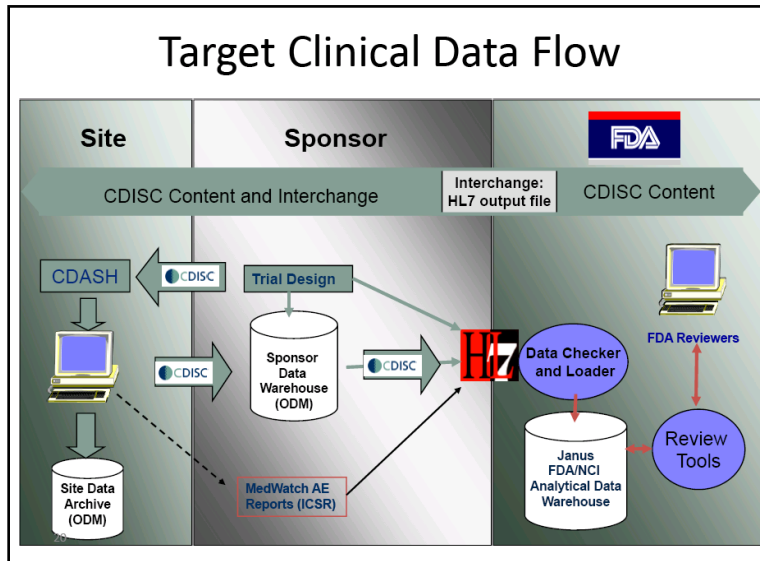


Figure 5 FDA PDUFA IV Clinical Data Flow

As mentioned earlier, the FDA currently accepts the study data using SDTM and sends the data as SAS V5 XPT files. The study metadata must be sent using a CRT-DDS structured XML file.

In 2008, the FDA released its PDUFA IV Information Technology Plan in which it indicated that interchange format for study metadata and data submitted electronically would be changing. Custom HL7 V3 (XML) messages are in the process of being developed to fulfill this task.

When adopted, the new messages will replace the SAS V5 XPT format and the CRT-DDS XML format. Figure 5 illustrates this point.

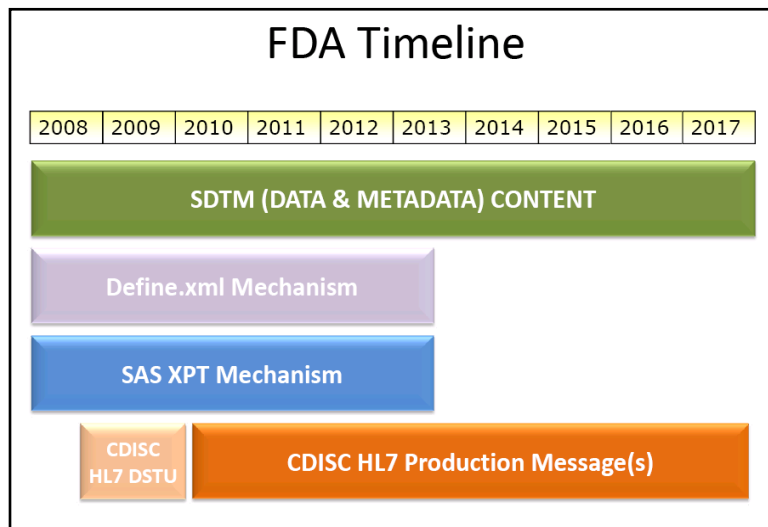


Figure 6 CDISC Interpretation of Submission Standards Evolution

The timing for this changeover is open to some interpretation. Figure 6 shows the CDISC group's thoughts as of late 2008.

The SDTM standard will still be used to scope the data and metadata going forward in the new HL7 messages. After the HL7 messages are officially adopted, there will be an overlap during which sponsors are expected to change over and finally the older formats will no longer be allowed.

The SAS Clinical Standards Toolkit was designed as a modular system, able to adapt to new standards and versions of those standards. It already has a flexible approach to the creation of XML files and moving to support CDISC-HL7 messages is planned.

CONCLUSION

The paper has shown that the SAS Clinical Standards Toolkit has been designed with the existing and (known) future requirements for standards in mind. It is flexible enough to handle a sponsor's custom requirements, allowing these to execute within a standard framework.

REFERENCES

- FDA. PDUFA IV Information Technology Plan. Available at <http://www.fda.gov/OHRMS/DOCKETS/98fr/FDA-2008-N-0352-bkg.pdf>.
- Vaughn, Carol. "Loading SDTM Data into the Janus Data Warehouse." Available at http://www.cdisc.org/publications/international_interchange_08/Session3B/CarolVaughn-Presentation.pdf.
- Iberson-Hurst, David. "CDISC Technical Strategy Into 2009." Available at http://www.cdisc.org/publications/international_interchange_08/Session1_Reports/20081028_TechnicalStrategy.pdf.
- Kilhullen, Michael. *Best Practices for Working with CDISC Metadata in the SAS Data Integration Server*. SAS Institute Inc. Available at <http://www.lexjansen.com/pharmasug/2007/sas/sa03.pdf>.
- Villiers, Peter. "SAS, CDISC and Clinical Data Integration." SAS Institute Inc. Available at <http://www.lexjansen.com/pharmasug/2008/sas/sa11.pdf>.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Peter Villiers
SAS Institute Inc.
SAS Campus Drive
Cary, NC, 27519, USA

Phone: +1 (919) 531-9083

Peter.Villiers@sas.com
<http://www.sas.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Table 1 Example SASReferences Data Set

Standard	Standard Version	Type	SubType	SASref	RefType	Path	order	memname	comment
CDISC-SDTM	3.1.1	autocall		code1	fileref		1		Std-specific macros
CST-FRAMEWORK	1.2	autocall		code2	fileref		2		Framework macros
CDISC-SDTM	3.1.1	control	reference	control	libref	&studyRootPath\control	.	sasreferences.sas7bdat	
CDISC-SDTM	3.1.1	control	validation	control	libref	&studyRootPath\control	.	validation_control.sas7bdat	
CDISC-SDTM	3.1.1	fmtsearch		srcfmt	libref	&studyRootPath\terminology\formats	1	formats.sas7bcat	Std/study-specific formats
CDISC-TERMINOLOGY		fmtsearch		cstfmt	libref		2		Global Library formats
CDISC-SDTM	3.1.1	lookup		refcntl	libref		.		
CDISC-SDTM	3.1.1	messages		sdtmsg	libref		1	messages.sas7bdat	
CST-FRAMEWORK	1.2	messages		cstmsg	libref		2		
CDISC-SDTM	3.1.1	properties	validation	valprop	fileref	&studyRootPath\programs	3	validation.properties	Properties specific to CDISC-SDTM validation
CUSTOM		referencecterm		ctref	libref	C:\Playpen\Validation\Custom\ControlledTerminology\coding-dictionaries	1	meddra.sas7bdat	Customers dictionary location
CDISC-SDTM	3.1.1	referencemetadata	column	refmeta	libref		.		
CDISC-SDTM	3.1.1	referencemetadata	table	refmeta	libref		.		
CDISC-SDTM	3.1.1	results	validationmetrics	results	libref	&studyRootPath\results	.	validation_metrics.sas7bdat	
CDISC-SDTM	3.1.1	results	validationresults	results	libref	&studyRootPath\results	.	validation_results.sas7bdat	
CDISC-SDTM	3.1.1	resultspackage	log	saslog	fileref	&studyRootPath\results	.	CSTprocess.log	
CDISC-SDTM	3.1.1	resultspackage	xml	package	fileref	&studyRootPath\results	.	resultspackage.xml	
CDISC-SDTM	3.1.1	sourcedata		srcdata	libref	&studyRootPath\data	.		Path to study-specific SDTM domain data sets
CDISC-SDTM	3.1.1	sourcemetadata	column	srcmeta	libref	&studyRootPath\metadata	.	source_columns.sas7bdat	Source of study SDTM column metadata
CDISC-SDTM	3.1.1	sourcemetadata	table	srcmeta	libref	&studyRootPath\metadata	.	source_tables.sas7bdat	Source of study SDTM table metadata

Table 2 SDTM Validation Control Table

Check no	Check id	standard	Standard version	Check source	Source id	Check severity	Check type	CodeSource	Use source metadata	Table scope	Column scope	...	Report all
201	SDTM0602	CDISC-SDTM	***	SAS	SAS0007	Warning	Multirecord	sdtmcheck notunique	Y	ALL			Y
202	SDTM0603	CDISC-SDTM	***	JanusFR	IR4004	Error	Multirecord	sdtmcheck notunique	Y	ALL	**SEQ		Y
209	SDTM0622	CDISC-SDTM	***	SAS	SAS0025	Warning	Multirecord	sdtmcheck notunique	Y	FINDINGS	[**TEST] [**TESTCD]		N
213	SDTM0641	CDISC-SDTM	***	JanusFR	R4005	Error	Multirecord	sdtmcheck notunique	Y	DM			Y
226	SDTM0651	CDISC-SDTM	***	JanusFR	R4140	Error	Multirecord	sdtmcheck notunique	Y	DV	[DVTERM] [DVDECOD]		Y
228	SDTM0661	CDISC-SDTM	***	JanusFR	R4083	Error	Multirecord	sdtmcheck notunique	Y	SUPP**			Y
230	SDTM0671	CDISC-SDTM	***	SAS	SAS0032	Warning	Multirecord	sdtmcheck notunique	Y	TS	[TSPARM] [TSPARMCD]		Y

Table 3 SDTM Validation Results Data Set

CheckId	Result Seq	Seq No	SrcData	resultid	Message	Result Severity	Result Flag	_cst_rc	Actual	Key Values
SDTM0001	1	1	SRCDATA.AE	CST0100	No errors detected in SRCDATA.AE	Info	0	0		
SDTM0001	1	2	SRCDATA.DM	CST0100	No errors detected in SRCDATA.DM	Info	0	0		
SDTM0001	1	3	SRCDATA.DS	CST0100	No errors detected in SRCDATA.DS	Info	0	0		
SDTM0001	1	4	SRCDATA.DV	CST0100	No errors detected in SRCDATA.DV	Info	0	0		
SDTM0001	1	5	SRCDATA.IE	CST0100	No errors detected in SRCDATA.IE	Info	0	0		
SDTM0001	1	6	SRCDATA.LB	CST0100	No errors detected in SRCDATA.LB	Info	0	0		
SDTM0001	1	7	SRCDATA.MH	CST0100	No errors detected in SRCDATA.MH	Info	0	0		
SDTM0001	1	8	SRCDATA.PF	CST0100	No errors detected in SRCDATA.PF	Info	0	0		
SDTM0001	1	9	SRCDATA.RELREC	SDTM0001	Domain SRCDATA.RELREC contains 0 observations	Warning	1	0		
SDTM0001	1	10	SRCDATA.SUPPAE	CST0100	No errors detected in SRCDATA.SUPPAE	Info	0	0		
SDTM0001	1	11	SRCDATA.SUPPALL	CST0100	No errors detected in SRCDATA.SUPPALL	Info	0	0		
SDTM0001	1	12	SRCDATA.SUPPDM	CST0100	No errors detected in SRCDATA.SUPPDM	Info	0	0		
SDTM0001	1	13	SRCDATA.SUPPQUAL	CST0100	No errors detected in SRCDATA.SUPPQUAL	Info	0	0		
SDTM0001	1	14	SRCDATA.SV	CST0100	No errors detected in SRCDATA.SV	Info	0	0		
SDTM0001	1	15	SRCDATA.TA	CST0100	No errors detected in SRCDATA.TA	Info	0	0		
SDTM0001	1	16	SRCDATA.TS	CST0100	No errors detected in SRCDATA.TS	Info	0	0		
SDTM0001	1	17	SRCDATA.VS	CST0100	No errors detected in SRCDATA.VS	Info	0	0		
SDTM0002	1	1	srcdata.DM	CST0100	No errors detected in srcdata.DM	Info	0	0		
SDTM0002	1	2	srcdata.EX	CST0003	Check not run - srcdata.EX could not be found	Error	-1	0		
SDTM0003	1	1	srcdata.DM	CST0100	No errors detected in srcdata.DM	Info	0	0		

Table 4 SDTM Validation Metrics

MetricParameter	RecCount	ResultId	SrcData	ResultSeq
# of domains tested	17	SDTM0001	ALL	1
# of observations	7	SDTM0001	SRCDATA.AE	1
# of observations	14	SDTM0001	SRCDATA.DM	1
# of observations	18	SDTM0001	SRCDATA.DS	1
# of observations	2	SDTM0001	SRCDATA.DV	1
# of observations	6	SDTM0001	SRCDATA.IE	1
# of observations	8	SDTM0001	SRCDATA.LB	1
# of observations	7	SDTM0001	SRCDATA.MH	1
# of observations	21	SDTM0001	SRCDATA.PF	1
# of observations	0	SDTM0001	SRCDATA.RELREC	1
# of observations	3	SDTM0001	SRCDATA.SUPPAE	1
# of observations	7	SDTM0001	SRCDATA.SUPPALL	1
# of observations	4	SDTM0001	SRCDATA.SUPPDM	1
# of observations	7	SDTM0001	SRCDATA.SUPPQUAL	1
# of observations	24	SDTM0001	SRCDATA.SV	1
# of observations	12	SDTM0001	SRCDATA.TA	1
# of observations	2	SDTM0001	SRCDATA.TS	1
# of observations	14	SDTM0001	SRCDATA.VS	1
# of domains tested	3	SDTM0002	DM+DS+EX	1
# of observations	14	SDTM0002	srcdata.DM	1
# of domains tested	1	SDTM0003	DM	1
# of observations	14	SDTM0003	srcdata.DM	1
# of distinct check invocations	4	METRICS	sdtm_validate	1
Errors (severity=High) reported	1	METRICS	sdtm_validate	1
Warnings (severity=Medium) reported	1	METRICS	sdtm_validate	1
Notes (severity=Low) reported	0	METRICS	sdtm_validate	1

Table 5 Example CRT-DDS SASReferences Data Set

Standard	Standard Version	Type	SubType	SASref	RefType	Path	order	memname	comment
CDISC-CRTDDS	1.0	autocall		code1	fileref		1		Model-specific macros
CST-FRAMEWORK	1.2	autocall		code2	fileref		2		Framework macros
CDISC-CRTDDS	1.0	refstylesheet	default	refxslt	fileref				Use a default style sheet
CDISC-CRTDDS	1.0	messages		crtddsmsg	libref		1		
CST-FRAMEWORK	1.2	messages		cstmsg	libref		2		
CDISC-CRTDDS	1.0	properties	write	writeprop	fileref		1		Set default write properties
CDISC-CRTDDS	1.0	external	stylesheet	extxslt	fileref	&studyRootPath\crtdds-output			Output directory for style sheet
CDISC-CRTDDS	1.0	external	xml	extxml	libref	&studyRootPath\crtdds-output		define.xml	Output location for the XML file.
CDISC-CRTDDS	1.0	results	writeresults	results	libref	&studyRootPath\results		crtdds_write_results	
CDISC-CRTDDS	1.0	sourcedata		srcdata	libref	&studyRootPath\crtdds-data			
CDISC-CRTDDS	1.0	sourcemetadate	column	srcmeta	libref	&studyRootPath\crtdds-metadata		source_columns	
CDISC-CRTDDS	1.0	sourcemetadate	table	srcmeta	libref	&studyRootPath\crtdds-metadata		source_tables	