

CLINICAL STANDARDS TOOLKIT 1.5

HOW DO I KNOW MY METADATA IS RIGHT?

GENE LIGHTFOOT, SAS, CARY, NC, US



- In the context of Clinical Standards Toolkit
- Information about the Toolkit
- Information about CDISC Models (SDTM, ODM, etc.)
- Information about Customer Standards
- Information about where things are.

- Validation is an industry requirement
- Customers want and require this
- Good clinical programming practice – affects clinical data
- Want to ensure product is behaving the way SAS intended.

- No systematic assessment of the validity of Clinical Standards Toolkit metadata.
- Code undergoes separate SAS testing before it is shipped
- Running the cstutil_checkds framework macro assesses structure and content
- SAS provides an IQ/OQ document that customer's can use.
Problem: IQ/OQ is a baseline created by SAS, sometimes the customer installation is a little different, causing some mismatches.

SAS Health & Life Sciences R&D and Product Management saw importance of providing a mechanism (process) for Internal Validation of the Clinical Standards Toolkit.

Objectives:

- To ensure SAS supplied metadata files are consistent and correct
- To provide functionality to customers to help validate custom standards

3 Areas to validate:

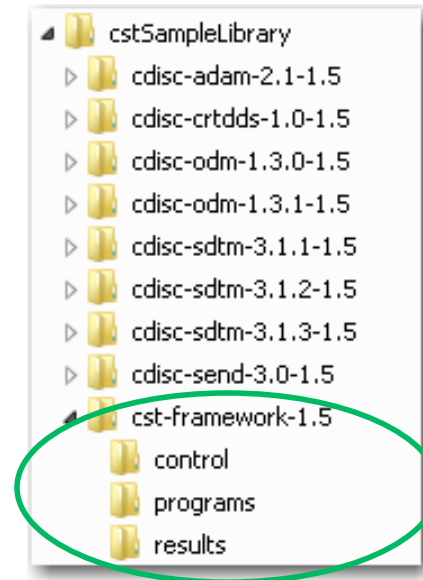
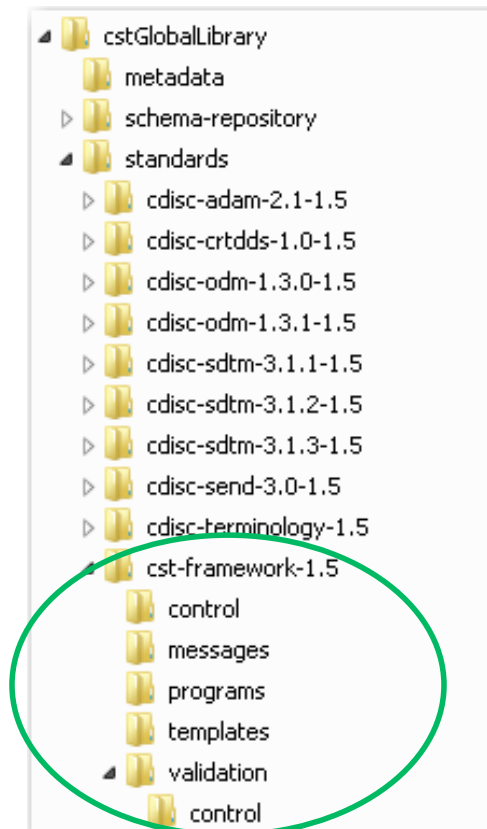
- Global and sample library
 - Framework and sample studies
- Standard reference metadata
 - CDISC standards
- Standard source metadata
 - Study level

Range of functionality

- Support IQ/OQ assessment and reporting
- Support registration of a new standard and/or updates to an existing standard
- Assess metadata consistency across files
- Assess structural validity of a metadata file
- Assess content of a metadata file
- Validate a SASReferences data set
- Evaluate validation check metadata
- Perform a process pre-check
- Perform run-time process validation
- Evaluate code module header compliance

- Clinical Standards Toolkit is standard/model based.
- Load information about Clinical Standards Toolkit as a Standard
- Populate the Metadata files
- Create validation_master data set containing internal validation checks.
- Use the Framework validation macros to validate itself

- Clinical Standards Toolkit cstGlobalLibrary and cstSampleLibrary folder structure created for Framework Standard



- CST_FRAMEWORK is registered (standards.sas7bdat)
- Standard supports validation is Y

	Name of standard	Mnemonic for standard	Root path for the standard	(Sample) study root path
10	CST-FRAMEWORK	CST	&_cstGRoot./standards/cst-framework-1.5	&_cstSRoot./cst-framework-1.5

Is this the default version for the standard [Y/N]?	Is this standard part of the CST framework [Y/N]?	Is this a data standard [Y/N]?	Standard supports validation [Y/N]?	Is this an xml-based standard [Y/N]?	The revision of the standard-standardversion
Y	Y	N	Y	N	1.5

Reference (Standard) Metadata is created dynamically for internal validation:

- reference_tables and reference_columns
- Different from other standards which are static
- More flexible for internal validation since framework metadata can change as new standards and/or customized standards are added.

Toolkit uses global macro variables extensively. Introduction of new metadata tables allows us validate these macros.

Standardmacrovariables data set – contains metadata about all macro **variables** used throughout Clinical Standards Toolkit

	Macro variable name	Macro variable label	Macro description	Is macro variable value required? (Y/N)	Is macrovalue case sensitive? (Y/N)	Value type(ANY,DISCRETE,INTEGER,DATASET)	Source file for macro variable
1	_cstCheckSortOrder	Specifies the order in which validation checks are to be run	This variable enables specification of the order in which the checks are to be run. The _DATA_ value indicates that checks are to be processed in the order defined in the Validation Control data set. You can specify a set of space-delimited keys from Validation Control columns (for example, checksource checkid).	N	N	ANY	validation
2	_cstColumnMetadat	Data set containing column-level metadata supporting validation	Data set that is used during processing that contains column-level metadata (derived from either the reference or study column metadata) that is used by the process.	Y	N	DATASET	validation
3	_cstDebug	Turns debugging on or off for the session	If on, then _cstDebugOptions are set. Many files remain in the Work library at process conclusion. Note that when _cstDebug=1, the size of the SAS log is significantly larger.	Y	N	DISCRETE	initialize
4	_cstDebugOptions	SAS session debugging options when _cstDebug=1	SAS system options set when _cstDebug=1.	N	N	ANY	initialize

Standardmacrovariabledetails data set – contains metadata about all macro variable **values** used throughout Clinical Standards Toolkit.

	Macro variable name	Macro variable value	Macro variable value label	Is this value the default? (Y/N)
1	_cstCheckSortOrder	_DATA_	Use order defined in the Validation Control data set	Y
2	_cstColumnMetadata	work._cstcolumnmetadata		Y
3	_cstDebug	0	Off	Y
4	_cstDebug	1	On	N
5	_cstDebugOptions	mprint mlogic symbolgen mautolocdisplay		Y
6	_cstFMTLibraries			Y
7	_cstMessageOrder	APPEND	Append records from multiple message data sets	Y
8	_cstMessageOrder	MERGE	Merge records from multiple message data sets	N

Messages data set – contains Framework Specific messages used by validation and reporting processes. Prefaced by CSTxxxx and CSTVxxx

	Result identifier	Standar version	Source of check	Record identifier used by checksource	Severity of check	Message text	Basis or explanation for result
81	CST0201	***	CST	CST0201	Warning	&_cstParm1	
82	CST0202	***	CST	CST0202	Error	&_cstParm1	
83	CSTV001	***	CST	CSTV001	Warning	Multiple records detected for standard &_cstParm1	
84	CSTV002	***	CST	CSTV002	Error	Standard is defined as XML but location of import XSL is missing	Example: XML standard without XSL/schema paths
85	CSTV003	***	CST	CSTV003	Error	Standard is defined as XML but location of export XSL is missing	Example: XML standard without XSL/schema paths

Addition of new /Templates folder containing metadata data sets. Used to validate data set structure. Each standard has its own set of templates.

 messages.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 metrics.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 results.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 sasreferences.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 standardlookup.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 standardmacrovariabledetails.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 standardmacrovariables.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 standards.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 standardsasreferences.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 validation_master.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB
 validation_stdref.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	17 KB

Validation_master – contains ALL checks created by SAS to validate that the installation is correct and required metadata is in place.

	Validation check identifier	Severity of check	SAS macro module name	Domains/data sets to which check applies	Columns to which check applies	Code logic used within code
1	CSTV001	Warning	cstcheck_notunique	glmeta.standards	standard+mnemonic	
2	CSTV002	Error	cstcheck_columncompare	glmeta.standards	[isxmlstandard][importxml]	data work._cstProblems;set &_cstDSName;if (upcase(&_cstColumn1) eq "Y" and &_cstColumn2 eq "");run;
3	CSTV003	Error	cstcheck_columncompare	glmeta.standards	[isxmlstandard][exportxml]	data work._cstProblems;set &_cstDSName;if (upcase(&_cstColumn1) eq "Y" and &_cstColumn2 eq "");run;
4	CSTV004	Error	cstcheck_columncompare	glmeta.standards	[isxmlstandard][schema]	data work._cstProblems;set &_cstDSName;if (upcase(&_cstColumn1) eq "Y" and &_cstColumn2 eq "");run;
5	CSTV005	Error	cstcheck_columncompare	glmeta.standards	[isxmlstandard][importxml]	data work._cstProblems;set &_cstDSName;if (upcase(&_cstColumn1) ne "Y" and &_cstColumn2 ne "");run;
6	CSTV006	Error	cstcheck_columncompare	glmeta.standards	[isxmlstandard][exportxml]	data work._cstProblems;set &_cstDSName;if (upcase(&_cstColumn1) ne "Y" and &_cstColumn2 ne "");run;
7	CSTV007	Error	cstcheck_columncompare	glmeta.standards	[isxmlstandard][schema]	data work._cstProblems;set &_cstDSName;if (upcase(&_cstColumn1) ne "Y" and &_cstColumn2 ne "");run;

New metadata columns have been added to handle Internal Validation

- IOType - (input, output, or both) determines if a file must be present and have r/w access.
- Filetype – Folder, dataset, catalog, or file
- Allowoverwrite – Can we write to the file (Y/N)
- Studylibraryrootpath – root directory for study library
- ...

New SAS macros have been written

- Cstutilcheckwriteaccess – If an output file can we write to the folder
- Cstutilfindvalidfile – Does the required file exist?
- ...

A sample library has been created for Framework validation containing framework specific programs and data.

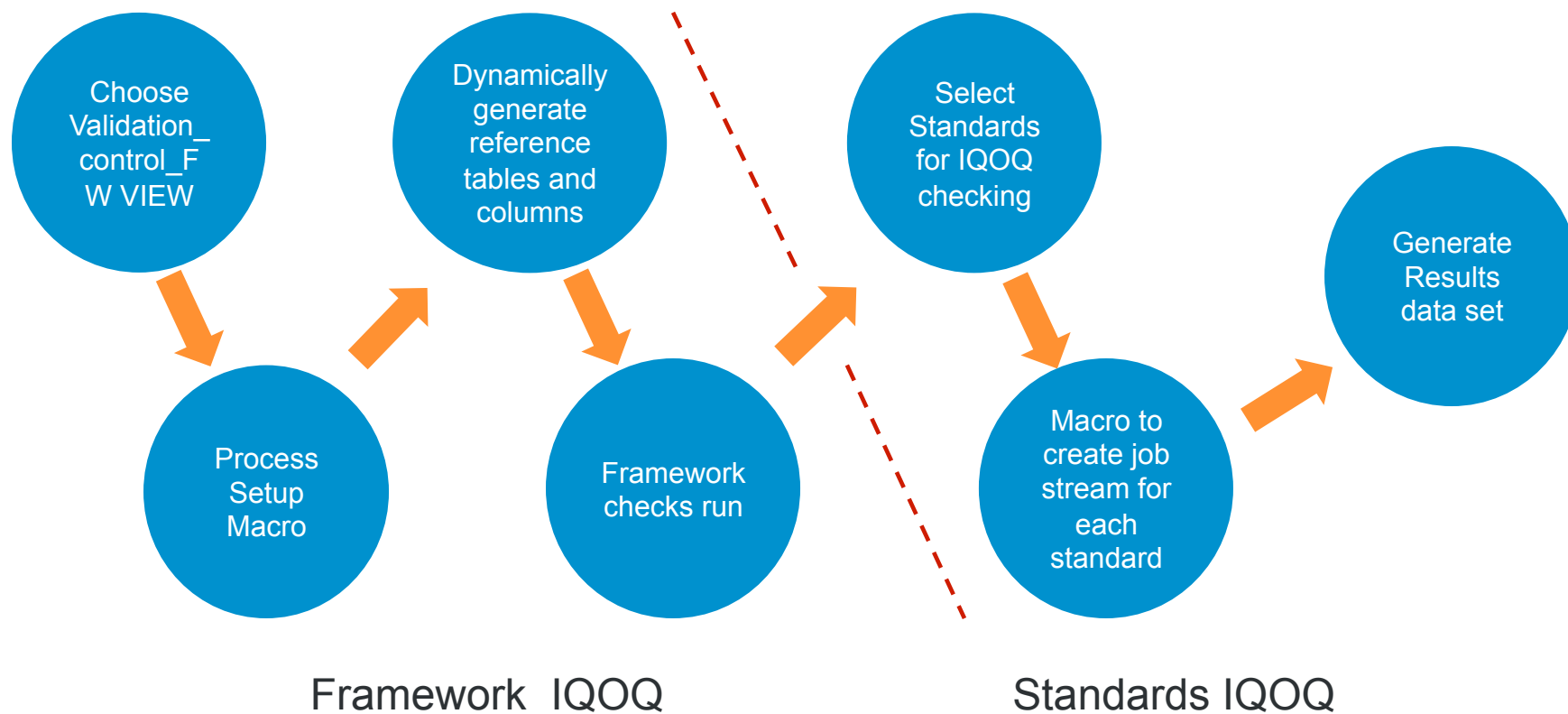
Driver programs

- validate_data.sas – run everything specified in validation_control
- validate_framework – framework specific checks.
- validate_iqoq – IQ/OQ specific checks.
- validate_standard – standard specific metadata checks

- Validation_control – Usually a copy or subset of validation_master allowing the user to select their checks.
- New to 1.5 – Validation_control VIEWS subset by checktype from validation_master:
 - FW-framework
 - STD – standard
 - STDIQOQ – standard iq/oq

	validation_control.sas7bdat	9/28/2012 3:27 PM	SAS Data Set	321 KB
	validation_control_fw.sas7bview	9/28/2012 3:27 PM	SAS Data Set View	5 KB
	validation_control_std.sas7bview	9/28/2012 3:27 PM	SAS Data Set View	5 KB
	validation_control_stdqiq.sas7bview	9/28/2012 3:27 PM	SAS Data Set View	5 KB

VALIDATE_IQOQ



Results are stored in the validation results data set

- Accessible by programmers
- Can be used in reports

Result identifier	Source data	Resolved message text from message file	Result severity (e.g., warning, error)	Problem detected? (0=no, otherwise yes)	Process status (Non-zero, aborted)	Actual value observed	Record-level keys + values
CSTV260	REFMETA.REFERENCE_TABLES	Required column cannot be null	Error	1	0	keys=	SASref=REFDATA,table=ItemGroupLeafTitles
CSTV260	REFMETA.REFERENCE_TABLES	Required column cannot be null	Error	1	0	keys=	SASref=REFDATA,table=MUTranslatedText
CST0100	REFMETA.REFERENCE_COLUMNS	No errors detected in REFMETA.REFERENCE_COLUMNS	Info	0	0		
CST0004	CSTCHECK_COLUMN	No columns evaluated - check validation_control specification	Warning: Check not run	-1	0	tableScope=gl_cntl.st	

Internal Validation by the Numbers*

- ~ 121 Internal Validation checks
- 15 new Clinical Standards Toolkit macros
- 8 new Framework metadata columns
- Number of new metadata tables
 - Framework – 2
 - Templates – 49 (currently across 9 standards)

*Numbers may vary as development was ongoing when slide was produced.

- SAS Clinical Standards Toolkit information can be found at:
 - <http://support.sas.com/rnd/base/cdisc/cst/index.html>
- SAS Knowledge Base/Focus Area
 - <http://support.sas.com/rnd/base/cdisc/cst/index.html>
- Availability of updates will generally be posted to the SAS and Clinical Trials Community, available at:
 - http://communities.sas.com/community/sas_and_clinical_trials

Please come see me in the SAS booth
Talk to your sales representative
Let SAS Technical Support know.
Contact me through e-mail

Questions?

THANK YOU



Contact information:

Gene Lightfoot
+1-919-531-3103
gene.lightfoot@sas.com
(Clinical Standards Toolkit Development)

Bill J. Gibson
+1-919-531-0996
billj.gibson@sas.com
(Product Management)