PhUSE 2011

Paper TS03

Implementing ADaM Using SAS Clinical Standards Toolkit 1.4®

Gene Lightfoot, SAS Institute, Cary, North Carolina, USA

ABSTRACT

This paper introduces the steps taken to add a new data standard, in this case ADaM 2.1, into SAS Clinical Standards Toolkit. By following the steps and processes defined here, users of the SAS Clinical Standards Toolkit to will be able to implement their own data standard.

INTRODUCTION

The ADaM Implementation Guide Version 1.0 released in December 2009 contains the latest information from the Clinical Data Information Standards Consortium (CDISC) for ADaM 2.1. These guidelines are used to generate the metadata for the ADaM data sets. The SAS Clinical Standards Toolkit version 1.4 has implemented this model so that customers can validate their ADaM data sets against these standardized inputs. This paper will describe the process SAS used to implement the ADaM model in Toolkit.

WHAT IS THE SAS CLINICAL STANDARDS TOOLKIT?

The Clinical Standards Toolkit (Toolkit) is an open-source solution designed to support SAS programmers in the clinical research industry. Most of the code is either base SAS or SAS macro code and is shipped with SAS beginning with Version 9.2; for XML-based standards some JAVA and XSLT code is used. Toolkit is a SAS-based solution for SAS users and is designed for two primary purposes:

- To provide SAS-based implementations of evolving clinical data standards
- To provide a framework that exploits these standards to meet common clinical research analysis and submission requirements.

Currently, the Toolkit supports only standards developed by CDISC. However, there is nothing in the product design that limits support of other standards, and it is anticipated that both current users and future Toolkit releases will support non-CDISC clinically-related standards.

Given that study design, data collection, and analysis and submission requirements of each research protocol are unique, and that Toolkit provides libraries of open-source SAS macros and code, user customization is expected and encouraged.

INITIALIZATION OF A NEW STANDARD IN TOOLKIT

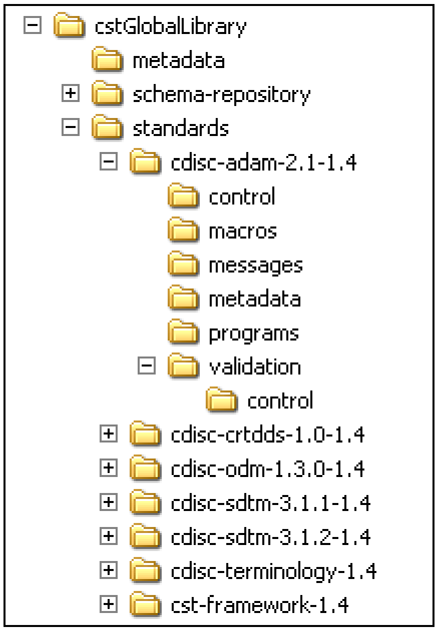
A new standard, whether it is a company standard, a CDISC standard, or Controlled Terminology has to be registered to take advantage of the functionality offered by the Toolkit. In order to register a standard to the Toolkit, the following three steps are required.

1. Create a folder structure in the Toolkit cstGlobalLibrary directory.
2. Create the required data sets needed by the Toolkit.
3. Register the standard to the Toolkit.

Each step is detailed below for the creation of the ADaM 2.1 data standard currently registered with Toolkit 1.4. These steps can be used to register any standard to the Toolkit. Currently there are three types of standards loaded into the Toolkit, each requiring their own unique requirements within Toolkit. These three types are Data (such as SDTM 3.1.2, ADaM 2.1), Controlled Terminology (as supplied by the National Cancer Institute (NCI) and can also include dictionaries such as MedDRA), and XML standards (such as ODM or CRT-DDS). Given the space constraints of this paper, it will only deal with implementing a data standard to the Toolkit.

For more detailed information about the ADaM model and its implementation, the reader is referred to the Toolkit 1.4 User documentation.

CREATION OF THE FOLDER STRUCTURE IN THE CSTGLOBALLIBRARY DIRECTORY



```
⊟ cstGlobalLibrary
    metadata
  ⊞ schema-repository
  ⊟ standards
    ⊟ cdisc-adam-2.1-1.4
        control
        macros
        messages
        metadata
        programs
      ⊟ validation
          control
    ⊞ cdisc-crtdds-1.0-1.4
    ⊞ cdisc-odm-1.3.0-1.4
    ⊞ cdisc-sdtm-3.1.1-1.4
    ⊞ cdisc-sdtm-3.1.2-1.4
    ⊞ cdisc-terminology-1.4
    ⊞ cst-framework-1.4
```

In a default installation of Toolkit on Windows, the Toolkit global library is installed under C:\cstGlobalLibrary. Under the standards folder are standard specific folders for each standard registered to the Toolkit, in the example to the left we see cdisc-adam-2.1-1.4, cdisc-crtdds-1.0-1,4, etc. These are naming conventions used by the Toolkit and are not needed for custom data standards. Under the standard specific folder for ADaM 2.1 the \control, \macros, \messages, \metadata, \programs, and \validation\control subfolders were created. Custom directories can be created for custom standards.

How do we know what is needed here? At a minimum the \control subfolder is required and contains the standards data set. Depending on information provided to the standards.sas7bdat data set about the standard will determine which folders and files are required in the Toolkit. See discussion below in “Creation of the Toolkit metadata/data files”.

The \control subfolder contains data sets specific to information about the standard. The \macros subfolder contains SAS macro programs developed specifically for the ADaM standard. The \messages subfolder contains the messages data set used for reporting and validation of the standard. The \metadata subfolder contains information specific to any data modeling used by the standard. For ADaM this subfolder contains SAS’ interpretation of the ADaM 2.1 Implementation Guide as provided by CDISC and is considered the “gold” standard for the ADaM model within Toolkit. The \programs folder contains any SAS programs used to register the model and any property files that are used by the Toolkit. The \validation\control subfolder contains the validation data for the standard and is used in the Toolkit validation process. As with the contents of the \metadata subfolder, this data is considered the “gold” standard for ADaM.

For initial data standard setup the user is required to create any needed folder structures. Once these folders are in place, they will need to be populated with the required data sets, macros, programs, and/or property files. Currently Toolkit does not provide a tool that creates and populates these for customized data standards.

CREATION OF THE TOOLKIT METADATA/DATA FILES

When initializing a data standard, what files are required by Toolkit? At a minimum, Toolkit requires the following data sets:

- standards
- standardsasreferences

Information provided to the standards.sas7bdat data set will determine the data sets and files needed to properly register a standard to the Toolkit. There are 5 columns in the standards data set that require a Y/N response, the values of three of these columns will determine the data set or file requirements for the Toolkit.

Is this the default version for the standard (Y/N)? [isstandarddefault]

> This value is used to set the standard as the default standard if more than one version of the standard exists.

Is this standard part of the CST Framework (Y/N)? [iscstframework]

> This value will always be N for a data standard

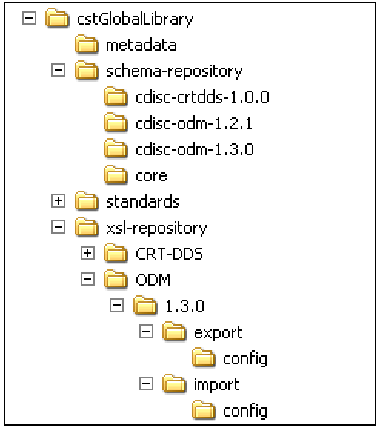
Is this a data standard (Y/N)? [isdatastandard]

> For any data standard (ie ADaM, SDTM, etc) this value is Y. When set to Y the \metadata subfolder is required. Within the \metadata folder the following data sets are required at a minimum: The reference_columns and reference_tables data sets. If the standard \programs, \messages, and \macros subfolders MAY be required if the user decides to use initialization files (\programs), write standard specific macro programs (\macros), and/or provide a standard specific messages data set for any user written reports (\messages). For ADaM 2.1 all of these criteria were met during its implementation.

Standard supports validation (Y/N)? [supportsvalidation]

> If the standard will undergo any form of automated validation within the Toolkit, this value needs to be set to Y. This requires the existence of the \validation\control and the \messages subfolders. ADaM 2.1 has a set of validations that are supplied by CDISC.

Is this an xml-based standard (Y/N)? [isxmlstandard]



```
⊟ cstGlobalLibrary
    metadata
  ⊟ schema-repository
      cdisc-crtdds-1.0.0
      cdisc-odm-1.2.1
      cdisc-odm-1.3.0
      core
  ⊞ standards
  ⊟ xsl-repository
    ⊞ CRT-DDS
    ⊟ ODM
      ⊟ 1.3.0
        ⊟ export
            config
        ⊟ import
            config
```

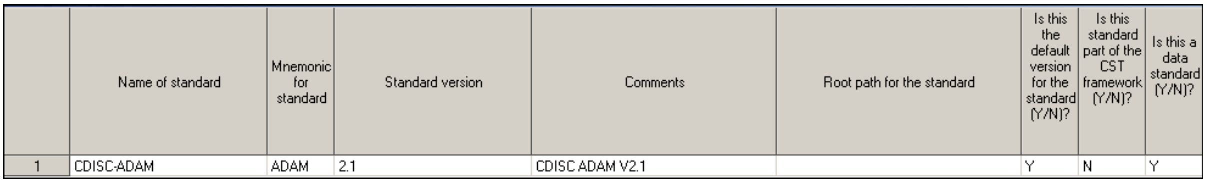
Since ADaM is not an XML based standard this is set to N. XML based standards such as ODM or CRT-DDS would be set to Y. A Y response here requires additional folder structure and the inclusion of XML transform code (*.xsl) and XML stylesheet information. In the figure at the left a corresponding standard, in this case ODM, would require a folder structure in the \schema-repository subfolder and a schema file. It would also require a corresponding set of \<standard>\<standard version> and corresponding \export and \import subfolders in the \xsl-repository subfolder. Any export or import transforms would be stored here.

XML based standards will not be discussed in this paper, this information is offered for reference only.

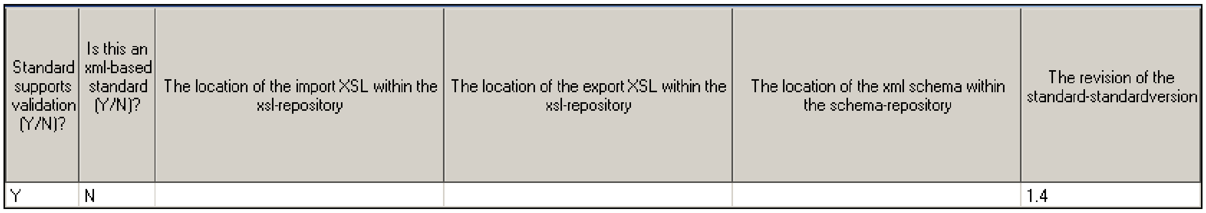
Following are the files shipped with Toolkit that were needed to implement the ADaM model. A brief description is included with each file. All columns are shown for each data set but not all of the content is shown. When implementing a customized data standard, the number of files may be different and the content will definitely be different. The files and metadata that constitute a standard vary from standard to standard. Use the files that define each standard, found in <global library>/standards/<standard> as templates, adding your customizations. You can rename or otherwise introduce your own subfolders and files that constitute each standard.

When creating these files for customized standards, it is suggested that the naming convention of the Toolkit tables and columns not be changed. Data sets and files created by the user to mimic those existing in Toolkit should follow structure as specified within the Toolkit to ensure proper function and facilitate easier implementation.

Standards.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\control) contains metadata about the ADaM model that is used by the Toolkit. This information is used to register a standard to the Toolkit. This is a one observation data set.

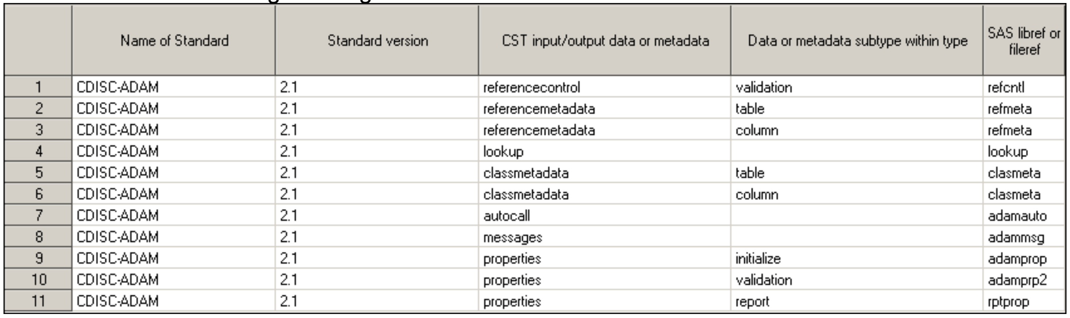


	Name of standard	Mnemonic for standard	Standard version	Comments	Root path for the standard	Is this the default version for the standard (Y/N)?	Is this standard part of the CST framework (Y/N)?	Is this a data standard (Y/N)?
1	CDISC-ADAM	ADAM	2.1	CDISC ADAM V2.1		Y	N	Y

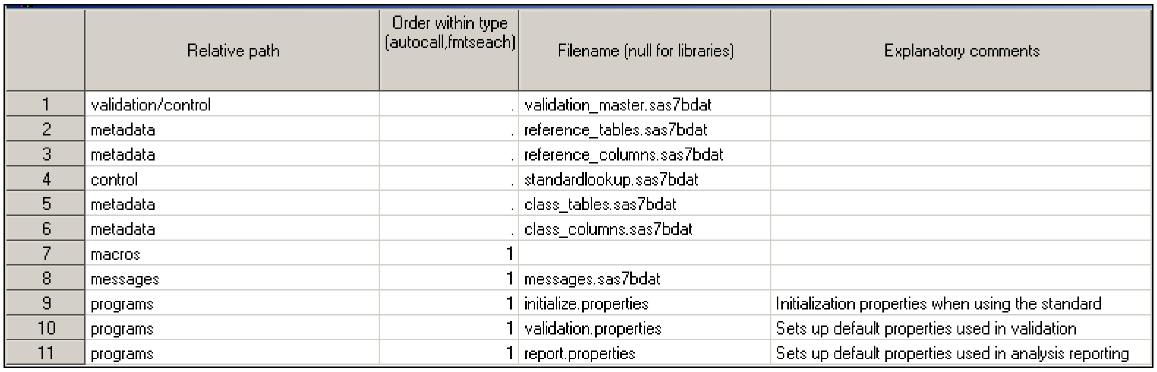


Standard supports validation (Y/N)?	Is this an xml-based standard (Y/N)?	The location of the import XSL within the xsl-repository	The location of the export XSL within the xsl-repository	The location of the xml schema within the schema-repository	The revision of the standard-standardversion
Y	N				1.4

Standardsasreferences.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\control) contains metadata telling Toolkit where things are located for the ADaM standard. This data set is used to standardize values in a sasreferences data set and is specific to the ADaM model registered to Toolkit. When running a process, if the sasreferences data set has blank values they will be populated with the values from this “standard” sasreferences. This allows standard ADaM settings to be global and shared across studies.

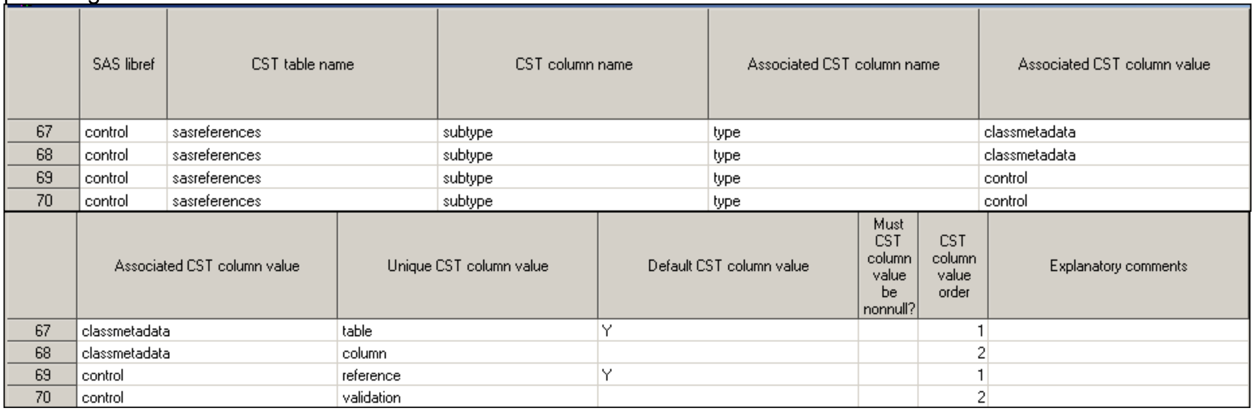


	Name of Standard	Standard version	CST input/output data or metadata	Data or metadata subtype within type	SAS libref or fileref
1	CDISC-ADAM	2.1	referencecontrol	validation	refcntl
2	CDISC-ADAM	2.1	referencemetadata	table	refmeta
3	CDISC-ADAM	2.1	referencemetadata	column	refmeta
4	CDISC-ADAM	2.1	lookup		lookup
5	CDISC-ADAM	2.1	classmetadata	table	clasmeta
6	CDISC-ADAM	2.1	classmetadata	column	clasmeta
7	CDISC-ADAM	2.1	autocall		adamauto
8	CDISC-ADAM	2.1	messages		adammsg
9	CDISC-ADAM	2.1	properties	initialize	adamprop
10	CDISC-ADAM	2.1	properties	validation	adamprp2
11	CDISC-ADAM	2.1	properties	report	rptprop



	Relative path	Order within type (autocall,fmtseach)	Filename (null for libraries)	Explanatory comments
1	validation/control	.	validation_master.sas7bdat	
2	metadata	.	reference_tables.sas7bdat	
3	metadata	.	reference_columns.sas7bdat	
4	control	.	standardlookup.sas7bdat	
5	metadata	.	class_tables.sas7bdat	
6	metadata	.	class_columns.sas7bdat	
7	macros	1		
8	messages	1	messages.sas7bdat	
9	programs	1	initialize.properties	Initialization properties when using the standard
10	programs	1	validation.properties	Sets up default properties used in validation
11	programs	1	report.properties	Sets up default properties used in analysis reporting

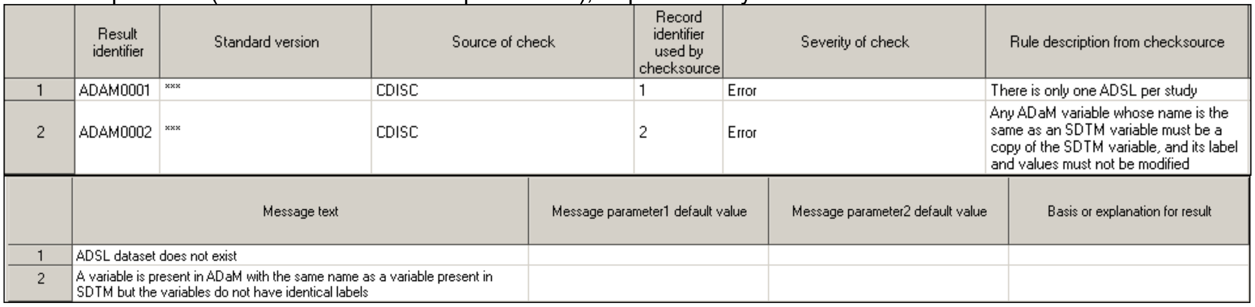
Standardlookup.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\control) provides a mechanism to capture valid values for discrete variables in the SAS Clinical Standards Toolkit metadata files. This data set supports such tasks as validating the content of the SAS Clinical Standards Toolkit metadata files and providing selectable values in the user interfaces of other tools and solutions.



	SAS libref	CST table name	CST column name	Associated CST column name	Associated CST column value
67	control	sasreferences	subtype	type	classmetadata
68	control	sasreferences	subtype	type	classmetadata
69	control	sasreferences	subtype	type	control
70	control	sasreferences	subtype	type	control

	Associated CST column value	Unique CST column value	Default CST column value	Must CST column value be nonnull?	CST column value order	Explanatory comments
67	classmetadata	table	Y		1	
68	classmetadata	column			2	
69	control	reference	Y		1	
70	control	validation			2	

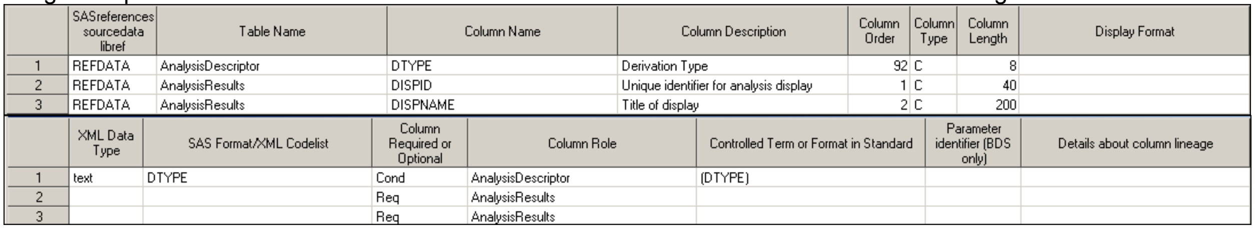
Messages.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\messages) contains the descriptive text used by the Toolkit validation process as well as the reporting process. This information can be gathered from several sources, such as any standard specific validation implementation guide, public domain validation products (such as WebSDM or OpenCDISC), or provided by the user.



	Result identifier	Standard version	Source of check	Record identifier used by checksource	Severity of check	Rule description from checksource
1	ADAM0001	***	CDISC	1	Error	There is only one ADSL per study
2	ADAM0002	***	CDISC	2	Error	Any ADaM variable whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label and values must not be modified

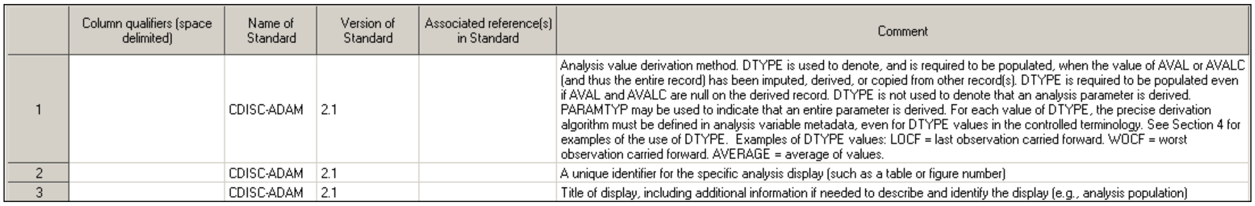
	Message text	Message parameter1 default value	Message parameter2 default value	Basis or explanation for result
1	ADSL dataset does not exist			
2	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels			

Class_columns.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\metadata) is provided by Toolkit as an input to the SAS Clinical Data Integration® product. Currently it is not needed in the base functionality provided by Toolkit. Information from this data set is used to populate drop down windows in the Clinical Data Integration product. This data set would not be needed for a customized data standard running in Toolkit.



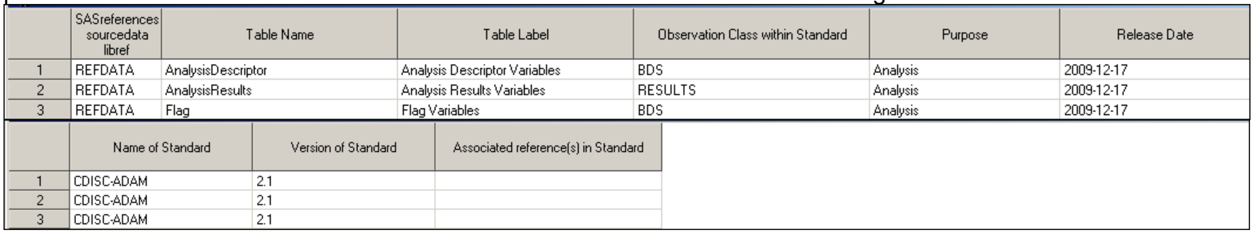
	SASreferences sourcedata libref	Table Name	Column Name	Column Description	Column Order	Column Type	Column Length	Display Format
1	REFDATA	AnalysisDescriptor	DTYPE	Derivation Type	92	C	8	
2	REFDATA	AnalysisResults	DISPID	Unique identifier for analysis display	1	C	40	
3	REFDATA	AnalysisResults	DISPNAME	Title of display	2	C	200	

	XML Data Type	SAS Format/XML Codelist	Column Required or Optional	Column Role	Controlled Term or Format in Standard	Parameter identifier (BDS only)	Details about column lineage
1	text	DTYPE	Cond	AnalysisDescriptor	(DTYPE)		
2			Req	AnalysisResults			
3			Req	AnalysisResults			



	Column qualifiers (space delimited)	Name of Standard	Version of Standard	Associated reference(s) in Standard	Comment
1		CDISC-ADAM	2.1		Analysis value derivation method. DTYPE is used to denote, and is required to be populated, when the value of AVAL or AVALC (and thus the entire record) has been imputed, derived, or copied from other record(s). DTYPE is required to be populated even if AVAL and AVALC are null on the derived record. DTYPE is not used to denote that an analysis parameter is derived. PARAMTYP may be used to indicate that an entire parameter is derived. For each value of DTYPE, the precise derivation algorithm must be defined in analysis variable metadata, even for DTYPE values in the controlled terminology. See Section 4 for examples of the use of DTYPE. Examples of DTYPE values: LOCF = last observation carried forward. WOCF = worst observation carried forward. AVERAGE = average of values.
2		CDISC-ADAM	2.1		A unique identifier for the specific analysis display (such as a table or figure number)
3		CDISC-ADAM	2.1		Title of display, including additional information if needed to describe and identify the display (e.g., analysis population)

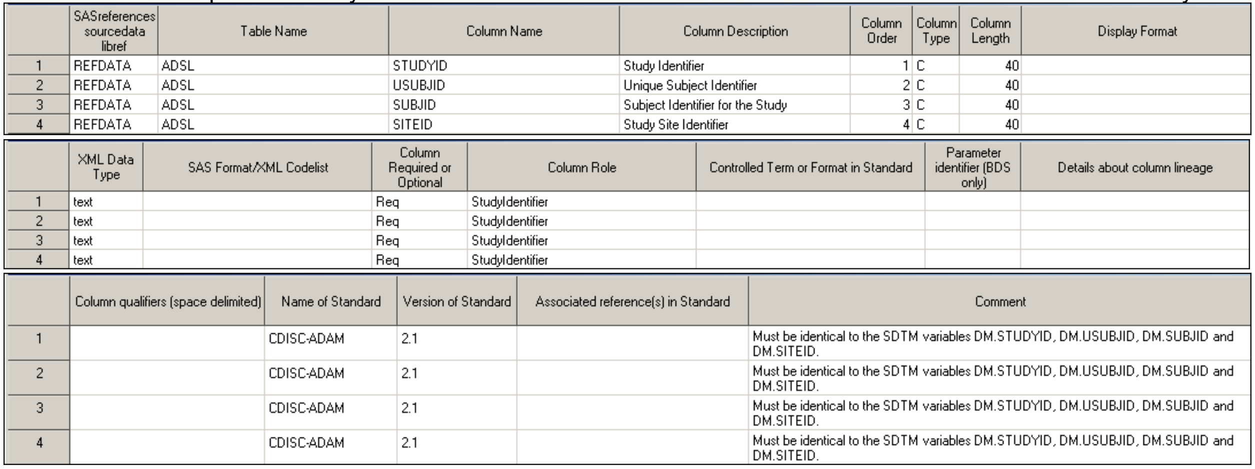
Class_tables.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\metadata) is provided by Toolkit as an input to the SAS Clinical Data Integration® product. Currently it is not needed in the base functionality provided by Toolkit. Information from this data set is used to populate drop down windows in the Clinical Data Integration product. This data set would not be needed for a customized data standard running in Toolkit.



	SASreferences sourcedata libref	Table Name	Table Label	Observation Class within Standard	Purpose	Release Date
1	REFDATA	AnalysisDescriptor	Analysis Descriptor Variables	BDS	Analysis	2009-12-17
2	REFDATA	AnalysisResults	Analysis Results Variables	RESULTS	Analysis	2009-12-17
3	REFDATA	Flag	Flag Variables	BDS	Analysis	2009-12-17

	Name of Standard	Version of Standard	Associated reference(s) in Standard
1	CDISC-ADAM	2.1	
2	CDISC-ADAM	2.1	
3	CDISC-ADAM	2.1	

References_columns.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\metadata) contains metadata about all of the columns registered in the ADaM standard within Toolkit. This information is derived from the ADaM Implementation Guide (V2.1) and is the SAS interpretation of the ADaM model. This information can be modified by the user if required. An example may be the need to change the length of a column. Reference_columns is considered "the gold standard" and, as such, only one should exist for each standard. Many of the ADaM validations will compare the study data metadata back to the reference standard metadata to validate conformity.

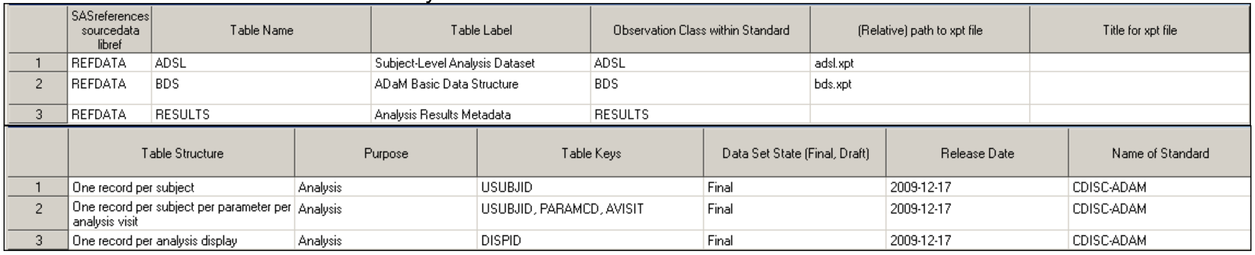


	SASreferences sourcedata libref	Table Name	Column Name	Column Description	Column Order	Column Type	Column Length	Display Format
1	REFDATA	ADSL	STUDYID	Study Identifier	1	C	40	
2	REFDATA	ADSL	USUBJID	Unique Subject Identifier	2	C	40	
3	REFDATA	ADSL	SUBJID	Subject Identifier for the Study	3	C	40	
4	REFDATA	ADSL	SITEID	Study Site Identifier	4	C	40	

	XML Data Type	SAS Format/XML Codelist	Column Required or Optional	Column Role	Controlled Term or Format in Standard	Parameter identifier (BDS only)	Details about column lineage
1	text		Req	StudyIdentifier			
2	text		Req	StudyIdentifier			
3	text		Req	StudyIdentifier			
4	text		Req	StudyIdentifier			

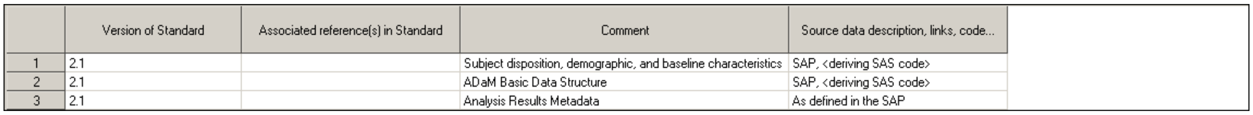
	Column qualifiers (space delimited)	Name of Standard	Version of Standard	Associated reference(s) in Standard	Comment
1		CDISC-ADAM	2.1		Must be identical to the SDTM variables DM.STUDYID, DM.USUBJID, DM.SUBJID and DM.SITEID.
2		CDISC-ADAM	2.1		Must be identical to the SDTM variables DM.STUDYID, DM.USUBJID, DM.SUBJID and DM.SITEID.
3		CDISC-ADAM	2.1		Must be identical to the SDTM variables DM.STUDYID, DM.USUBJID, DM.SUBJID and DM.SITEID.
4		CDISC-ADAM	2.1		Must be identical to the SDTM variables DM.STUDYID, DM.USUBJID, DM.SUBJID and DM.SITEID.

Reference_tables.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\metadata) contains metadata about all of the tables registered in the ADaM standard within Toolkit. This information is derived from the ADaM Implementation Guide (V2.1) and is the SAS interpretation of the ADaM model. Currently there are three tables defined in reference_tables: ADSL (ADaM Subject Level), BDS (Basic Data Structure), and RESULTS. This information can be modified by the user if required. Reference_tables is considered "the gold standard" and only one should exist for each standard. Many of the validations will compare the study data metadata back to the reference standard metadata to validate conformity.



	SASreferences sourcedata libref	Table Name	Table Label	Observation Class within Standard	(Relative) path to xpt file	Title for xpt file
1	REFDATA	ADSL	Subject-Level Analysis Dataset	ADSL	adsl.xpt	
2	REFDATA	BDS	ADaM Basic Data Structure	BDS	bds.xpt	
3	REFDATA	RESULTS	Analysis Results Metadata	RESULTS		

	Table Structure	Purpose	Table Keys	Data Set State (Final, Draft)	Release Date	Name of Standard
1	One record per subject	Analysis	USUBJID	Final	2009-12-17	CDISC-ADAM
2	One record per subject per parameter per analysis visit	Analysis	USUBJID, PARAMCD, AVISIT	Final	2009-12-17	CDISC-ADAM
3	One record per analysis display	Analysis	DISPID	Final	2009-12-17	CDISC-ADAM



	Version of Standard	Associated reference(s) in Standard	Comment	Source data description, links, code...
1	2.1		Subject disposition, demographic, and baseline characteristics	SAP, <deriving SAS code>
2	2.1		ADaM Basic Data Structure	SAP, <deriving SAS code>
3	2.1		Analysis Results Metadata	As defined in the SAP

Validation_domainsbycheck.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\ validation\ control) contains records for each domain that is to be validated by each check in the Validation Master data set. This data set is used by the reporting tools that are provided with the SAS Clinical Standards Toolkit to report domain-specific errors. It is also available to other programs and applications that might need to subset checks that are applicable to specific domains. The version of the Validation_domainsbycheck data set that is supplied by SAS is built from the version of the Validation Master data set that is also supplied by SAS. If the tableScope and columnScope columns are modified, then the Validation_domainsbycheck data set must also be modified or rebuilt.

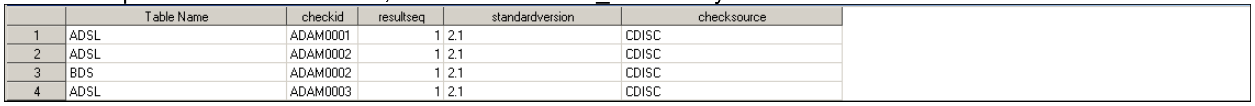
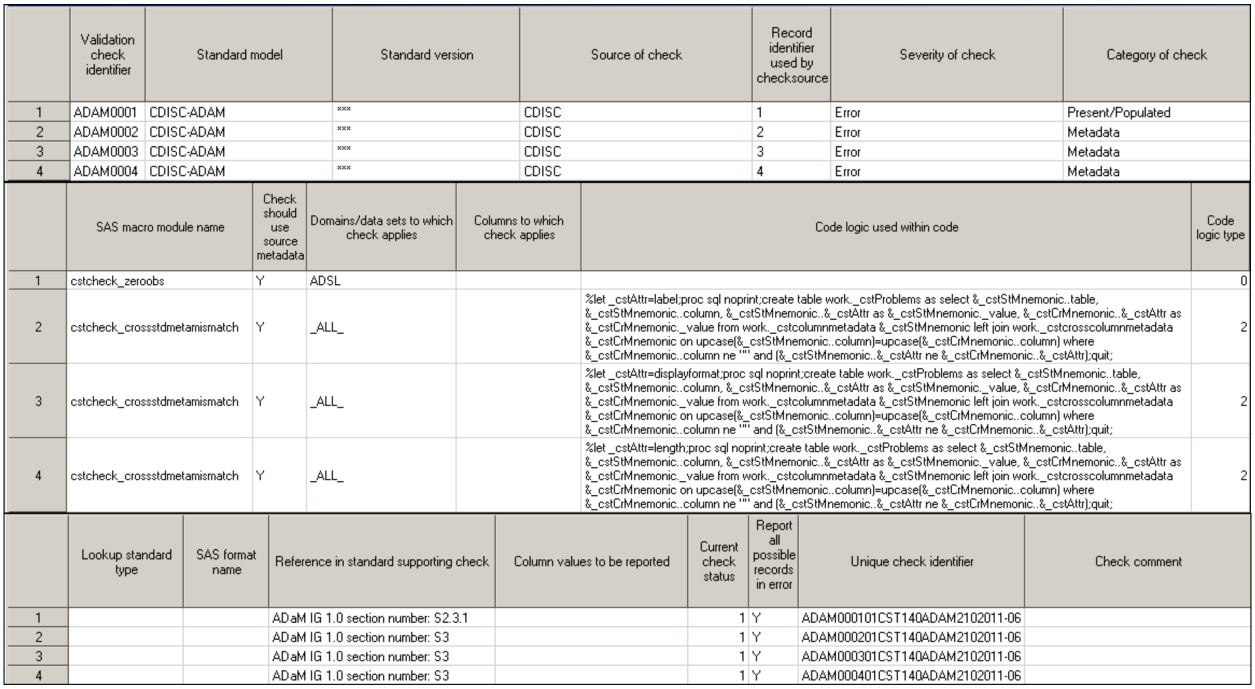


	Table Name	checkid	resultseq	standardversion	checksource
1	ADSL	ADAM0001	1	2.1	CDISC
2	ADSL	ADAM0002	1	2.1	CDISC
3	BDS	ADAM0002	1	2.1	CDISC
4	ADSL	ADAM0003	1	2.1	CDISC

Validation_master.sas7bdat (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\validation\control) contains information needed to implement the Toolkit validation process for ADam 2.1. This contains all of the validations that are available for the registered standard. Just like the reference_tables and reference_columns data sets, there should only be one copy of this file for each data standard. Some of the key elements in this data set include the "Validation check identifier" [checkid], a unique ID for each check and it is also used to retrieve the corresponding message from the messages data set. The "SAS macro module name" [codesource] provides the name of the SAS Toolkit validation macro used to perform the specific check. "Domains/data sets to which check applies" [tablescope] provides a list of the data sets being validated by the specific check (_ALL_ is a keyword used by Toolkit to specify that every data set is validated). "Column to which check applies" [columnscope] provides the columns being evaluated by the validation check. "Code logic used within code" [codelogic] provides SAS code for those macros that require it. Depending on the macro, this could be a SAS statement (ie a single line of code) or a complete data step (or proc sql statement). Information in this data set can also be modified to suit the needs of the user. Information used to populate this data set was provided by the CDISC ADaM Validation Checks Version 1.0 and Version 1.1 documents.

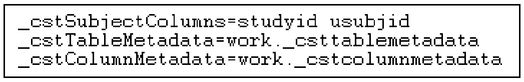


	Validation check identifier	Standard model	Standard version	Source of check	Record identifier used by checksource	Severity of check	Category of check
1	ADAM0001	CDISC-ADAM	***	CDISC	1	Error	Present/Populated
2	ADAM0002	CDISC-ADAM	***	CDISC	2	Error	Metadata
3	ADAM0003	CDISC-ADAM	***	CDISC	3	Error	Metadata
4	ADAM0004	CDISC-ADAM	***	CDISC	4	Error	Metadata

	SAS macro module name	Check should use source metadata	Domains/data sets to which check applies	Columns to which check applies	Code logic used within code	Code logic type
1	cstcheck_zeroobs	Y	ADSL			0
2	cstcheck_crossstdmetamismatch	Y	_ALL_		%let _cstAttr=label;proc sql noprint;create table work._cstProblems as select &_cstStMnemonic..table, &_cstStMnemonic..column, &_cstStMnemonic..&_cstAttr as &_cstStMnemonic._value, &_cstCrMnemonic..&_cstAttr as &_cstCrMnemonic._value from work._cstcolumnmetadata &_cstStMnemonic left join work._cstcrosscolumnmetadata &_cstCrMnemonic on upcase(&_cstStMnemonic..column)=upcase(&_cstCrMnemonic..column) where &_cstCrMnemonic..column ne "" and (&_cstStMnemonic..&_cstAttr ne &_cstCrMnemonic..&_cstAttr);quit;	2
3	cstcheck_crossstdmetamismatch	Y	_ALL_		%let _cstAttr=displayformat;proc sql noprint;create table work._cstProblems as select &_cstStMnemonic..table, &_cstStMnemonic..column, &_cstStMnemonic..&_cstAttr as &_cstStMnemonic._value, &_cstCrMnemonic..&_cstAttr as &_cstCrMnemonic._value from work._cstcolumnmetadata &_cstStMnemonic left join work._cstcrosscolumnmetadata &_cstCrMnemonic on upcase(&_cstStMnemonic..column)=upcase(&_cstCrMnemonic..column) where &_cstCrMnemonic..column ne "" and (&_cstStMnemonic..&_cstAttr ne &_cstCrMnemonic..&_cstAttr);quit;	2
4	cstcheck_crossstdmetamismatch	Y	_ALL_		%let _cstAttr=length;proc sql noprint;create table work._cstProblems as select &_cstStMnemonic..table, &_cstStMnemonic..column, &_cstStMnemonic..&_cstAttr as &_cstStMnemonic._value, &_cstCrMnemonic..&_cstAttr as &_cstCrMnemonic._value from work._cstcolumnmetadata &_cstStMnemonic left join work._cstcrosscolumnmetadata &_cstCrMnemonic on upcase(&_cstStMnemonic..column)=upcase(&_cstCrMnemonic..column) where &_cstCrMnemonic..column ne "" and (&_cstStMnemonic..&_cstAttr ne &_cstCrMnemonic..&_cstAttr);quit;	2

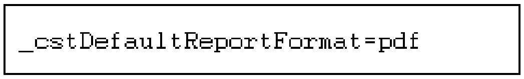
	Lookup standard type	SAS format name	Reference in standard supporting check	Column values to be reported	Current check status	Report all possible records in error	Unique check identifier	Check comment
1			ADaM IG 1.0 section number: S2.3.1		1	Y	ADAM000101CST140ADAM2102011-06	
2			ADaM IG 1.0 section number: S3		1	Y	ADAM000201CST140ADAM2102011-06	
3			ADaM IG 1.0 section number: S3		1	Y	ADAM000301CST140ADAM2102011-06	
4			ADaM IG 1.0 section number: S3		1	Y	ADAM000401CST140ADAM2102011-06	

Property files are used in the Toolkit to set values for standard level global macro variables used throughout the Toolkit. This is used to facilitate reuse and standardize information consumed by the Toolkit. Users can also modify or add to these values based on their needs. Currently there are 3 property files shipped with Toolkit for ADaM, initialize, validation, and report.



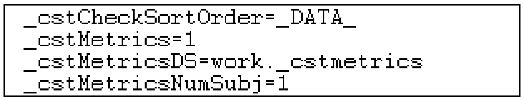
```
_cstSubjectColumns=studyid usubjid
_cstTableMetadata=work._csttablemetadata
_cstColumnMetadata=work._cstcolumnmetadata
```

Initialize.properties (stored in C:\cstGlobalLibrary\standards\ cdisc-adam-2.1-1.4\programs) contains generic information used to initially setup the standard in Toolkit. For ADaM 2.1 Toolkit sets the subject columns to uniquely identify subjects and sets the temporary table and column metadata files as work files.



```
_cstDefaultReportFormat=pdf
```

Report.properties (stored in C:\cstGlobalLibrary\standards\cdisc-adam-2.1-1.4\programs) contains a default value for the report file extension. Currently in Toolkit 1.4 only PDF, HTML, and CVS file types are supported.



```
_cstCheckSortOrder=_DATA_
_cstMetrics=1
_cstMetricsDS=work._cstmetrics
_cstMetricsNumSubj=1
```

Validation.properties (stored in C:\cstGlobalLibrary\standards\ cdisc-adam-2.1-1.4\programs) contains default values for the global macro variables used in the validation and reporting processes. For a more in depth review of this information please refer to the Clinical Standards Toolkit User documentation.

REGISTER THE STANDARD

After the directory hierarchy and the metadata files have been created and verified, the standard must now be registered to the Toolkit. ADaM 2.1 is shipped with SAS 9.3 and registered to the Toolkit during the installation process. The following steps are not needed for ADaM, but are presented here so the user has a more complete understanding of the process if a customized standard is being loaded into Toolkit. The following code defines and registers a new standard. The code can also be used to register a new version of an existing standard.

```
/*Step 1. Ensure that the macro variable pointing to the global standards library exists.*/

%cstutil_setcstgroot;

/*Step 2. Register the standard with the Toolkit global standards library*/

%cst_registerStandard( _cstRootPath=%nrstr(&_cstGRoot./standards/myStandard),
                       _cstControlSubPath=control,
                       _cstStdDSName=standards,
                       _cstStdSASRefsDSName=StandardSASReferences);
```

Step 1 ensures that the macro variable that contains the global standards library path is set. Step 2 registers the standard by passing the following information:

- The main path to the directory that contains the standard version's files.
- The path to the registration data sets that are used to populate the global standards library metadata data sets. This is the name of the subfolder in the _cstRootPath parameter value.
- The names of the Standards and StandardSASReferences data sets. These data sets have the same structure as the data sets in the global standards library metadata directory. Both of these data sets are required to define a new standard or a new version of a standard.

The _cstRootPath parameter uses %nrstr(&_cstGRoot) so that the &_cstGRoot is registered as a macro variable. This specification allows the global standards library to be moved or copied without reregistering the full path of the new standard.

ADAM SAMPLE STUDY

Sample ADaM data was derived from the SAS Toolkit SDTM 3.1.2 sample domains as well as the examples from the CDISC ADaM Pilot study conducted a few years back. In addition, mock tables were created so that the ADaM data could be manipulated to produce analysis tables as well as testing the “one proc away” intent of ADaM and these tables were used to produce the report metadata for ADaM. The ADaM data sets used with the Toolkit are ADSL, ADQS, and ADAE. These files are shipped with the ADaM sample study and are located at: C:\Program Files\SASHome\SASClinicalStandardsToolkitADaM21\1.4\sample\cdisc-adam-2.1\sascstdemodata\data.

- ADSL – Contains the subject level data and is based on the ADSL metadata supplied in the CDIS ADaM implementation guide.
- ADQS – Contains questionnaire data and is based on the BDS metadata as supplied by CDISC.

- ADAE – Contains adverse event data and is based on The ADaM Data Structure for Adverse Event Analysis Version 1.0 supplemental document release by CDISC on February 2, 2011.

Currently there are no tools supplied with the Toolkit to perform the creation of populated ADaM data sets, however these data sets can be generated using the SAS Clinical Data Integration® product.

ADAM VALIDATION

ADaM validation required the creation of a validation_master data set as described earlier in this paper. ADaM presented a unique problem to the developers at SAS. It was the first model implemented in the Toolkit that performed cross standard validation. In other words, the ADaM model required specific checks between metadata and data against the SDTM model. Since Toolkit validation is driven my SAS macros, this required the creation of "cross standard" validation macros. Earlier versions of the SAS Clinical Standards Toolkit did not need to address cross standards validation and, therefore, macros to handle these were not developed. With the release of the SAS Clinical Standards Toolkit 1.4, two macros were developed: cstcheck_crossstdcomparedomains.sas and cstcheck_crossstdmetamismatch.sas. These macros are located in !sasroot/cstframework/sasmacro.

cstcheck_crossstdcomparedomains.sas

The cstcheck_crossstdcomparedomains macro generally compares values for one or more columns in one table against either those same columns in another domain in another standard, or compares values against metadata from the comparison standard. The macro requires use of codelogic as a full DATA step or PROC SQL invocation. This DATA or SQL step assumes as input a work copy of the column metadata data set returned by the cstutil_buildcollist macro. Any resulting records in the derived data set represent errors to be reported.

Example validation checks that use this macro are these:

- ADaM subject not found in the SDTM dm domain
- ADaM SDTM domain reference (for traceability) but the SDTM domain is unknown.

An ADaM 2.1 validation check that uses this macro is ADAM0053 and the rule description for this check, taken from the CDISC ADaM Validation document, reads as follows: *"Invalid STUDYID/USUBJID combination not found in the SDTM Demographics domain."*

The message text for this check is this: *"The values of USUBJID are not present in SDTM.DM."*

Here is sample code from the codelogic field from the ADaM 2.1 validation_master data set for validation ADAM0053. In the example below, &_cstSQLColList and &_cstCrossDataLib are generated by the macro prior to execution of codelogic.

```
proc sql noprint;
  create table work._cstproblems as
    select &_cstSQLColList from &_cstDSName
    except select &_cstSQLColList from &_cstCrossDataLib..dm;
quit;
```

cstcheck_crossstdmetamismatch.sas

The cstcheck_crossstdmetamismatch macro identifies inconsistencies between metadata across registered standards. The macro requires use of _cstCodeLogic as a full DATA step or PROC SQL invocation. This DATA or SQL step assumes as input a work copy of the column metadata data set returned by the cstutil_buildcollist macro. Any resulting records in the derived data set represent errors to be reported.

Assumptions:

- No data content is accessed for this check.
- Both study and reference metadata are available to assess compliance.
- _cstProblems includes two columns (plus any others, as needed). The mnemonics come from the Global Library standards data set:
  - &_cstStMnemonic._value (For example, ADAM_value containing the value of the column of interest from the primary standard.)
  - &_cstCrMnemonic._value (For example, SDTM_value containing the value of the column of interest from the comparison standard.)

Required global macro variables:

- _cstcrossstd: The name of the comparison standard, this is also used as a parameter to initialize _cstCrMnemonic above.
- _cstcrossstdver: The version of the comparison standard
- _cstrunstd: The primary standard, this is also used as a parameter to initialize _cstStMnemonic above.
- _cstrunstdver: The version of the primary standard

An ADaM 2.1 validation check that uses this macro is ADAM0002 and the rule description for this check, taken from the CDISC ADaM Validation document reads as follows*: "Any ADaM variable whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label and values must not be modified."*

The message text for this check is: *"A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels."*

Here is sample code from the codelogic field from the ADaM 2.1 validation_master data set for validation ADAM0002. In the example below &_cstStMnemonic=ADAM and &_cstCrMnemonic=SDTM.

```
%let _cstAttr=label;
proc sql noprint;
  create table work._cstProblems as
    select &_cstStMnemonic..table, &_cstStMnemonic..column,
           &_cstStMnemonic..&_cstAttr as &_cstStMnemonic._value,
           &_cstCrMnemonic..&_cstAttr as &_cstCrMnemonic._value
  from work._cstcolumnmetadata &_cstStMnemonic left join
    work._cstcrosscolumnmetadata &_cstCrMnemonic
    on upcase(&_cstStMnemonic..column)=upcase(&_cstCrMnemonic..column)
    where &_cstCrMnemonic..column ne "" and
         (&_cstStMnemonic..&_cstAttr ne &_cstCrMnemonic..&_cstAttr);
quit;
```

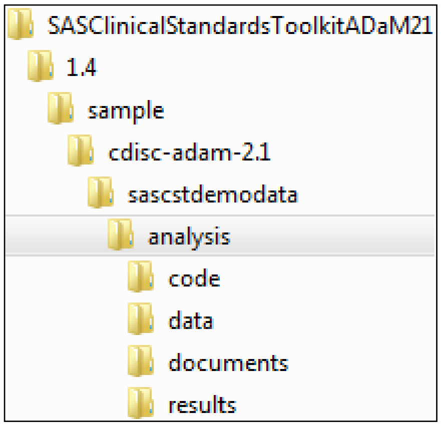
SAMPLE STUDY (CSR) REPORT ILLUSTRATING USE OF REPORT/RESULTS METADATA

The CDISC ADaM Document provides specifications for the structure and content of analysis data sets, as well as a suggested metadata format for documenting the analysis results generated that use the analysis data sets. Analysis results metadata describe the major attributes of a specified analysis result found in a clinical study report or submission, and support traceability from a result used in a statistical display to the data in the analysis data sets. The SAS Clinical Standards Toolkit representation of the ADaM standard includes a sample implementation of an analysis reporting methodology. The sample reporting methodology provided within the SAS Clinical Standards Toolkit is intended to be representative of similar industry methodologies. The intent is not to provide a definitive methodology, but to illustrate the interaction of reporting components through adoption of the ADaM standard.

Key Clinical Trial Reporting Components Addressed in Toolkit

Analysis Results (Tables, Figures, and Listings)	The set of statistical displays (for example, text, tabular or graphical presentation of results) or inferential statements such as p-values or estimates of treatment effect.
TFL Metadata (to include table shells)	Commonly provided as "table shells" which provide templates for the statistical displays. May also include display-specific metadata (often as Microsoft Excel files) used by the analysis programs to generate the displays.
Analysis Results Metadata	Defined by the Analysis Data Model Document, Section 5.3.
Analysis Programs	Programming code to use the analysis data sets (and optionally TFL metadata) to create the analysis results.

The majority of the files supporting the ADaM sample reporting methodology provided by the SAS Clinical Standards Toolkit reside in the ADaM analysis folder located at: !sasroot/../SASClinicalStandardsToolkitADaM21/1.4/sample/cdisc-adam-2.1/sascstdemodata/analysis.



- SASClinicalStandardsToolkitADaM21
  - 1.4
    - sample
      - cdisc-adam-2.1
        - sascstdemodata
          - analysis
            - code
            - data
            - documents
            - results

At left is an illustration of the ADaM Analysis folder hierarchy. Here are some of the noteworthy folders:

The **code** subfolder contains the code to create each statistical display. This corresponds to the Analysis Programs component described in the table above.

The **data** subfolder contains the display-specific metadata noted in the TFL Metadata component described in the table above.

The **documents** subfolder contains table shells for the TFL Metadata component.

The **results** folder contains several sample statistical displays, which correspond to the Analysis Results component.

TFL Metatatdata
A common industry reporting strategy is to create table shells, or templates, which specify the desired output for each statistical display. The SAS Clinical Standards Toolkit provides several sample table shells in this file: !sasroot/../SASClinicalStandardsToolkitADaM21/1.4/sample/cdisc-adam-2.1/sascstdemodata/analysis/documents/ Mock_tables_shells.pdf.

One of these displays, a table reporting patient demographics (Table 14.2.01), is presented here:

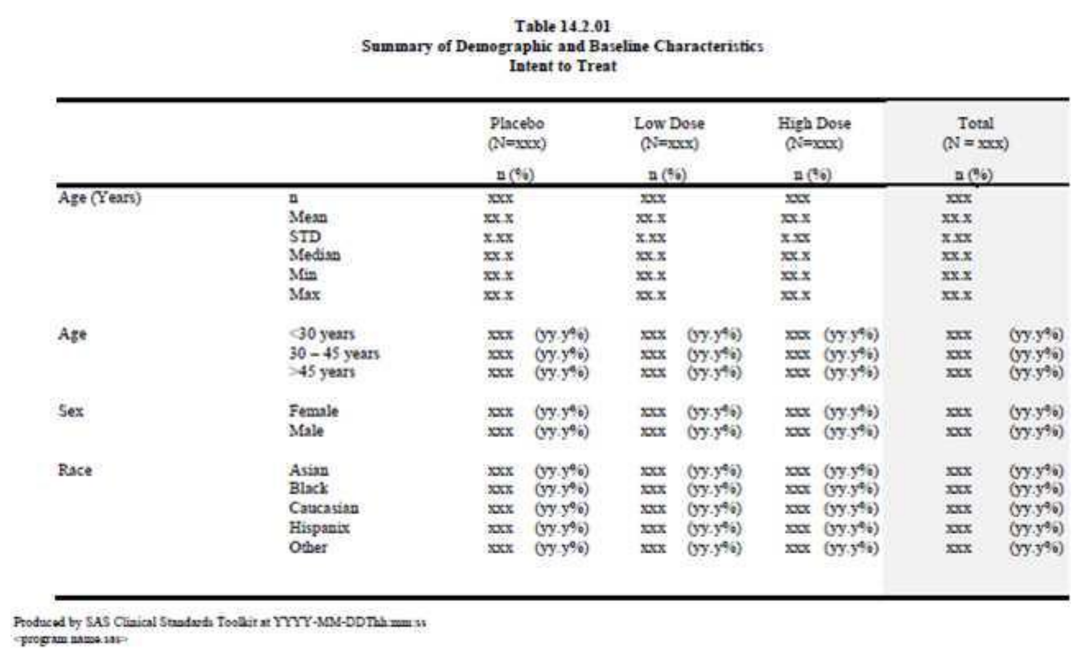


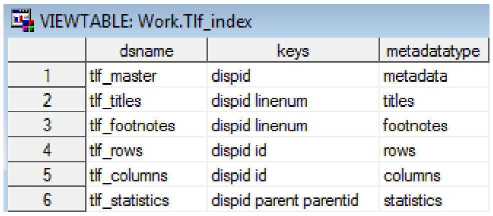
Table 14.2.01
Summary of Demographic and Baseline Characteristics
Intent to Treat

		Placebo (N=xxx)		Low Dose (N=xxx)		High Dose (N=xxx)		Total (N = xxx)	
		n (%)		n (%)		n (%)		n (%)	
Age (Years)	n	xxx		xxx		xxx		xxx	
	Mean	xx.x		xx.x		xx.x		xx.x	
	STD	x.xx		x.xx		x.xx		x.xx	
	Median	xx.x		xx.x		xx.x		xx.x	
	Min	xx.x		xx.x		xx.x		xx.x	
	Max	xx.x		xx.x		xx.x		xx.x	
Age	<30 years	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	30 – 45 years	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	>45 years	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
Sex	Female	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	Male	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
Race	Asian	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	Black	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	Caucasian	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	Hispanix	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)
	Other	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)	xxx	(yy.y%)

Produced by SAS Clinical Standards Toolkit at YYYY-MM-DDThh:mm:ss
<program name.sas>

The elements of each table shell (for example, titles, footnotes, headings, column and row labels, cell formatting, and so on) are sometimes captured in a metadata format, often in Microsoft Excel files. The usual intent is to create a set of reporting macros that can generate analysis reports based upon metadata, such that changes in metadata are all that are required to modify and rerun any report. For the SAS Clinical Standards Toolkit 1.4, a sample set of metadata is included that illustrates the use of such metadata within the ADaM reporting environment. This supplemental metadata is provided in this file: !sasroot/../SASClinicalStandardsToolkitADaM21/1.4/ sample/cdisc-adam-2.1/sascstdemodata/metadata/tlfddt.xml.

To interpret this metadata, a sample SAS XML map file (tlfddt.map) is provided in the same folder. SAS data sets, representing this XML metadata, are provided in the library of SAS files found in the folder located in: !sasroot/../ SASClinicalStandardsToolkitADaM21/1.4/sample/cdisc-adam-2.1/sascstdemodata/ analysis/data.

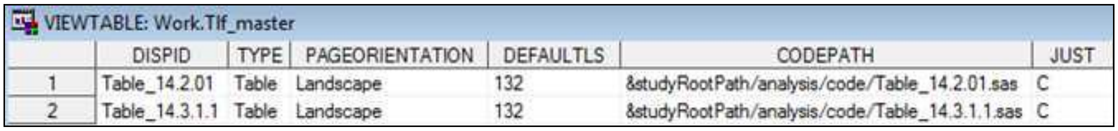


VIEWTABLE: Work.Tlf_index

	dsname	keys	metadatatype
1	tlf_master	dispid	metadata
2	tlf_titles	dispid linenum	titles
3	tlf_footnotes	dispid linenum	footnotes
4	tlf_rows	dispid id	rows
5	tlf_columns	dispid id	columns
6	tlf_statistics	dispid parent parentid	statistics

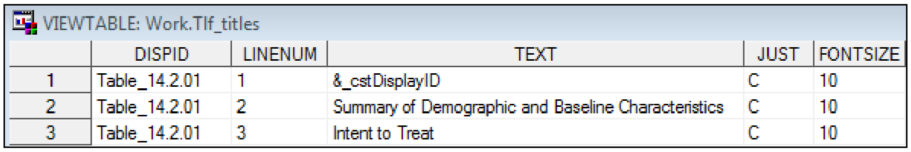
The following figures provide examples of some of the metadata available in the source XML files that have been extracted into SAS data sets. At left is the Tlf_index data set providing the data set names, the keys, and the metadata type associated with the table.

Pictured below is the Tlf_master data set. Row 1 of the Tlf_master data set describes a centered landscape table and where the generating code can be found. The titles for that table are provided in the Tfl_titles file displayed below Tlf_master. Note that these correspond to the Table Shell titles as specified in the table above.



VIEWTABLE: Work.Tlf_master

	DISPID	TYPE	PAGEORIENTATION	DEFAULTLS	CODEPATH	JUST
1	Table_14.2.01	Table	Landscape	132	&studyRootPath/analysis/code/Table_14.2.01.sas	C
2	Table_14.3.1.1	Table	Landscape	132	&studyRootPath/analysis/code/Table_14.3.1.1.sas	C



VIEWTABLE: Work.Tlf_titles

	DISPID	LINENUM	TEXT	JUST	FONTSIZE
1	Table_14.2.01	1	&_cstDisplayID	C	10
2	Table_14.2.01	2	Summary of Demographic and Baseline Characteristics	C	10
3	Table_14.2.01	3	Intent to Treat	C	10

Analysis Programs

The analysis program to generate the sample Table 14.2.01 is located in this folder: !sasroot/../ SASClinicalStandardsToolkitADaM21/1.4/sample/cdisc-adam-2.1/sascstdemodata/ analysis/code. Two versions are provided:

- Table_14.2.01.sas uses the TFL metadata described above.
- Table_14.2.01_nomd.sas does not rely on the presence of the TFL metadata to generate the report output.

As noted above, these sample analysis programs do not fully use the sample TFL metadata that has been defined and made available in the SAS Clinical Standards Toolkit. The basic coding strategy adopted with each SAS Clinical Standards Toolkit sample analysis program is to build each section (one or more row combinations) and to concatenate these sections into a single input file used by PROC REPORT.

A sample driver module is provided to perform the process setup, to define (or reference) the sasreferences data set, to perform any required report setup, and to call the generic ADaM reporting macro adam_createdisplay(). This driver located in this folder: !sasroot/../SASClinicalStandardsToolkitADaM21/1.4/sample/cdisc-adam2.1/sascstdemodata/ programs/analyze_data.sas. Within this driver, a call is made to adam_createdisplay() for each analysis report to be produced:

```
%adam_createdisplay
(displaysrc=Metadata,useanalysisresults=N,usetlfddt=Y,displayid=%str(Table_14.2.01));
```

To automate this process to create all analysis reports for a study, it would be necessary to cycle through any available metadata as listed in the Tlf_master data set above to construct multiple calls to the adam_createdisplay macro. The adam_createdisplay macro header provides an overview of the macro functionality and a summary of the defined macro parameters.

FUTURE GENERATION OF DEFINE.XML

As of the writing of this paper the define.xml specifications for ADaM are unpublished. SAS has created its own interpretation of the components of the define.xml for use with ODM and CRT-DDS and there is nothing in this design that would prevent the user from creating the ADaM define.xml. Once the specification is published, SAS will develop the necessary Toolkit modules to generate the define file.

CONCLUSION

SAS Clinical Toolkit 1.4 is a tool designed to implement, manage, and validate a company's standard. Its use of open SAS code in the form of macros, base SAS, and SAS data sets, allows users to customize and model the Toolkit to fit their needs. This paper addressed the thought process of SAS developers in implementing ADaM 2.1 and the new challenges presented by that model. New validation macros were developed to handle cross standard type validation checks and the introduction of reporting metadata was demonstrated within the product. Since the Toolkit is SAS based, users are encouraged to create their own macros and implement their own metadata within the product.

FURTHER READING

- CDISC Analysis Data Model, Version 2.1 available at www.cdisc.org/adam
- CDISC ADaM Implementation Guide, Version 1.0 available at www.cdisc.org/adam
- CDISC ADaM Basic Data Structure for Time-to-Event Analysis 1.0 at www.cdisc.org/adam
- The ADaM Data Structure for Adverse Event Analysis Version 1.0 Prepared by the CDISC Analysis Data Model Team (ADaM)
- SAS Clinical Standards Toolkit V1.4 User's Guide

CONTACT INFORMATION

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

Gene Lightfoot
SAS Institute Inc.
SAS Campus Drive R4318
Cary, North Carolina 27513 USA
+1 (919) 677-8000
+1 (919) 677-4444 (Fax)
gene.lightfoot@sas.com
www.sas.com