PharmaSUG 2012 - Paper PO12

Automatic Consistency Checking of Controlled Terminology and Value Level Metadata between ADaM Datasets and Define.xml for FDA Submission

Xiangchen (Bob) Cui, Vertex Pharmaceuticals, Cambridge, MA
Min Chen, Vertex Pharmaceuticals, Cambridge, MA

ABSTRACT

When submitting clinical study data (SDTM and ADaM data sets) in electronic format to the FDA, it is preferable to submit data definition tables (define.xml) and a reviewer guide (define.pdf). It is desirable to ensure the consistency between data sets and define files, and achieve technical accuracy and operational efficiency. This paper introduces a SAS® macro approach to automate consistency-checking of controlled terminology and value level metadata between ADaM data sets and define.xml. It avoids the waste of time and resources for verification of the consistency and/or resolution of inconsistency at a later stage. It also details how to develop ADaM Metadata (programming specification) for automation purposes, illustrates five scenarios of mismatches from consistency checking, and provides corresponding resolutions to these mismatches.

INTRODUCTION

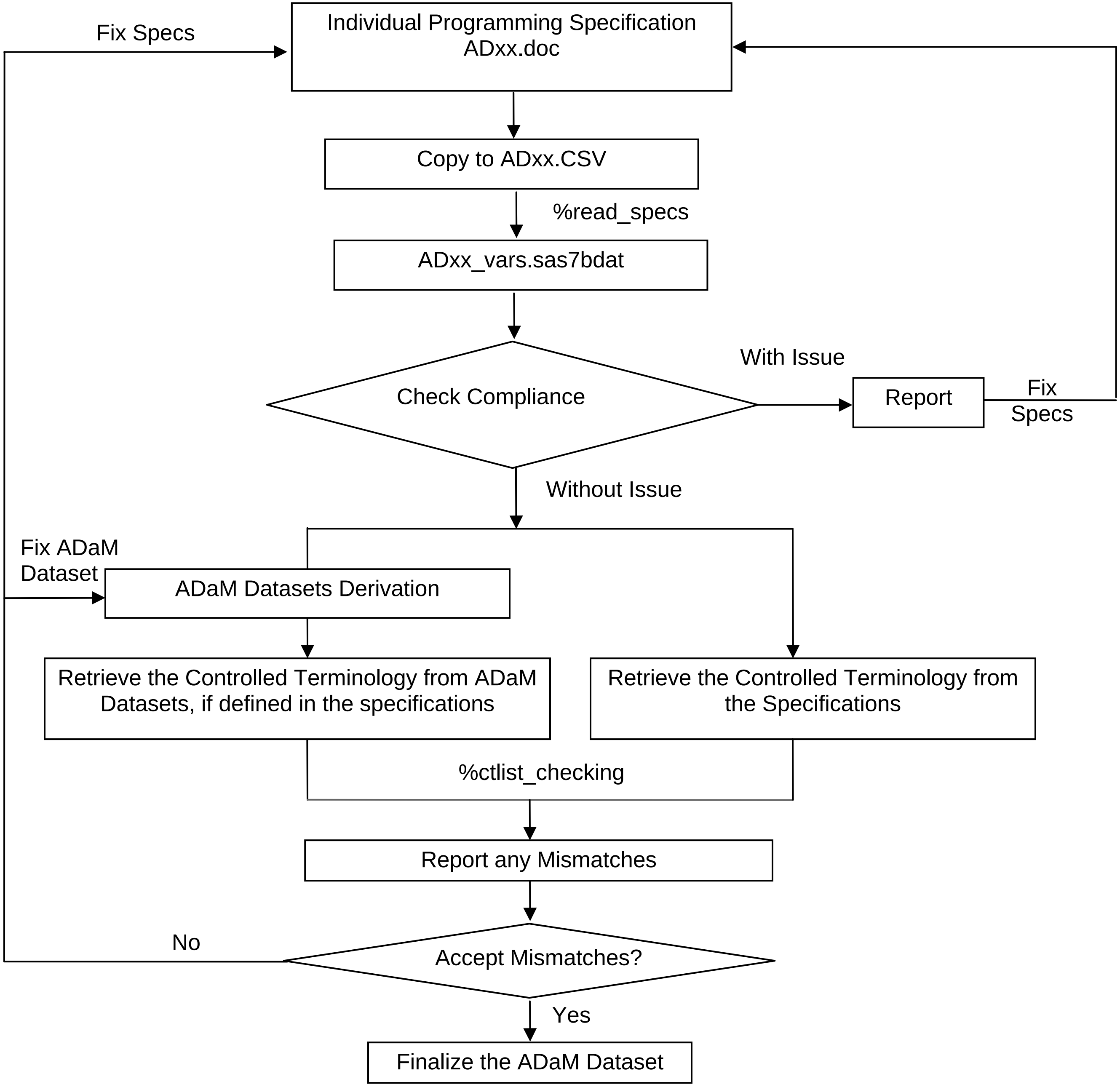
It is important to ensure that the define files are consistent with the datasets described within it for FDA submissions. The lack of consistency in many submissions has been documented [1]. We propose automatic consistency checking for controlled terminology between ADaM datasets and programming specifications as a solution to this regulatory concern.

The ADaM programming specifications are the unique source to manage metadata and are used to automatically generate define files. The ADaM datasets controlled terminology is described in the programming specifications. The controlled terminology in ADaM datasets is composed of value level metadata originating from source SDTM datasets for ADaM Basic Data Structure (BDS) Datasets, sponsor-defined terminology for the pair of corresponding variables from each dataset, controlled terminology inherited from SDTM domains, and therapeutic-specific terminology defined by FDA. The consistency of controlled terminology between ADaM datasets and the programming specifications ensures consistency between ADaM datasets and define files.

Based on the classification of controlled terminology in ADaM datasets, the guideline how to write ADaM programming specification for the controlled terminology in “Controlled Terms or Formats” column is introduced to make the automation of the consistency checking feasible. The specification for the controlled terminology provides the clear message to the programmers and FDA reviewers for the controlled terms, in addition to controlled terminology. A macro **%read_specs** is called to retrieve and store the information, including the controlled terminology information, from each programming specification for a variety of automation purposes in a SAS dataset (metadata), and a macro **%ctlist_checking** is called to compare the controlled terminology from the programming specification with the ones from ADaM datasets. It provides the summary reports about the mismatches if anything is detected. Five scenarios of mismatches are illustrated and their corresponding resolutions are provided in the paper to the readers as a reference.

Since the automation of consistency checking is conducted from the beginning of ADaM programming to the end for FDA submission, the high quality of the submissions can be achieved in a cost-effective and efficient way.

Display 1 shows the process flow.



Display 1. Overview of Process Flow

AN INTRODUCTION OF CONTROLLED TERMINOLOGY IN ADAM DATASETS

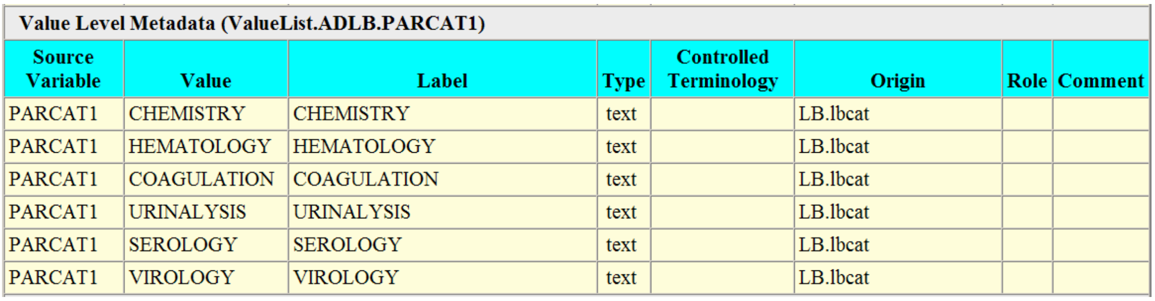
Controlled terminology represents a discrete set of values for a given variable. These sets of values for ADaM datasets may be value level metadata originated from source SDTM datasets to ADaM Basic Data Structure (BDS) Datasets, sponsor-defined terminology for a code-decode variable pair, controlled terminology inherited from SDTM domains, and therapeutic-specific terminology defined by FDA.

LIST OF FOUR KINDS OF CONTROLLED TERMINOLOGY

1. Value-Level Metadata for the ADaM Basic Data Structure (BDS) Datasets

Analysis parameter value-level metadata is required for all ADaM BDS datasets, which describes an analysis value within a given analysis parameter or a set of analysis parameters. The value list is originated from the SDTM Findings domain. The code list for the analysis parameters, PARCAT1 and PARAMCD, can help to determine the unique analysis parameter values in a dataset, and serve as an analysis parameter index and identifiers. Detailed information can be added when the analysis parameters are assigned to specific values.

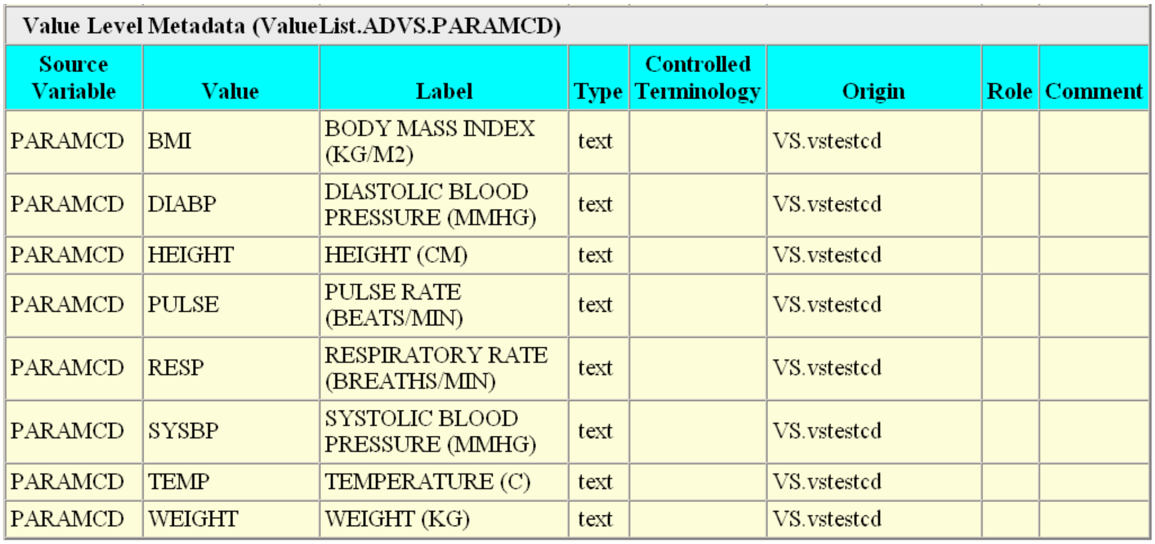
Examples of value-level metadata from the value level metadata section of our define.xml are shown in Display 2 and Display 3. Display 2 shows PARCAT1 in ADLB dataset. All possible values of the variable PARCAT1 in ADLB are listed in the Value Column, equal to the Label Column. Display 3 shows PARAMCD in ADVS. PARAMCD contains the short name of the analysis parameter in PARAM with 1:1 mapping between them. All possible values of PARAMCD and PARAM in ADVS can be found in the Value Column and the Label Column, respectively.



Value Level Metadata (ValueList.ADLB.PARCAT1)

Source Variable	Value	Label	Type	Controlled Terminology	Origin	Role	Comment
PARCAT1	CHEMISTRY	CHEMISTRY	text		LB.lbcat		
PARCAT1	HEMATOLOGY	HEMATOLOGY	text		LB.lbcat		
PARCAT1	COAGULATION	COAGULATION	text		LB.lbcat		
PARCAT1	URINALYSIS	URINALYSIS	text		LB.lbcat		
PARCAT1	SEROLOGY	SEROLOGY	text		LB.lbcat		
PARCAT1	VIROLOGY	VIROLOGY	text		LB.lbcat		

Display 2. An Example of Value-Level Metadata for PARCAT1



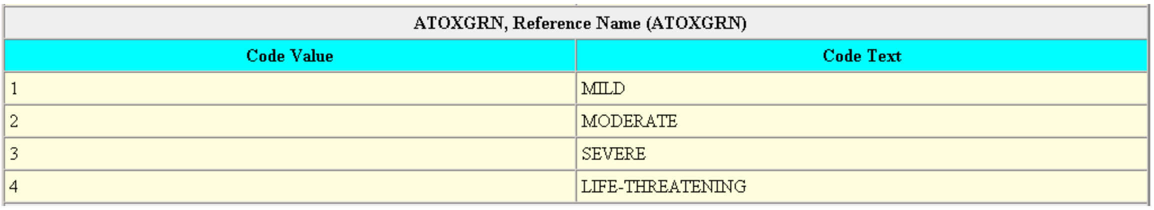
Value Level Metadata (ValueList.ADVS.PARAMCD)

Source Variable	Value	Label	Type	Controlled Terminology	Origin	Role	Comment
PARAMCD	BMI	BODY MASS INDEX (KG/M2)	text		VS.vstestcd		
PARAMCD	DIABP	DIASTOLIC BLOOD PRESSURE (MMHG)	text		VS.vstestcd		
PARAMCD	HEIGHT	HEIGHT (CM)	text		VS.vstestcd		
PARAMCD	PULSE	PULSE RATE (BEATS/MIN)	text		VS.vstestcd		
PARAMCD	RESP	RESPIRATORY RATE (BREATHS/MIN)	text		VS.vstestcd		
PARAMCD	SYSBP	SYSTOLIC BLOOD PRESSURE (MMHG)	text		VS.vstestcd		
PARAMCD	TEMP	TEMPERATURE (C)	text		VS.vstestcd		
PARAMCD	WEIGHT	WEIGHT (KG)	text		VS.vstestcd		

Display 3. An Example of Value-Level Metadata for PARAMCD

2. Sponsor-Defined Controlled Terminology for Decoding Purpose

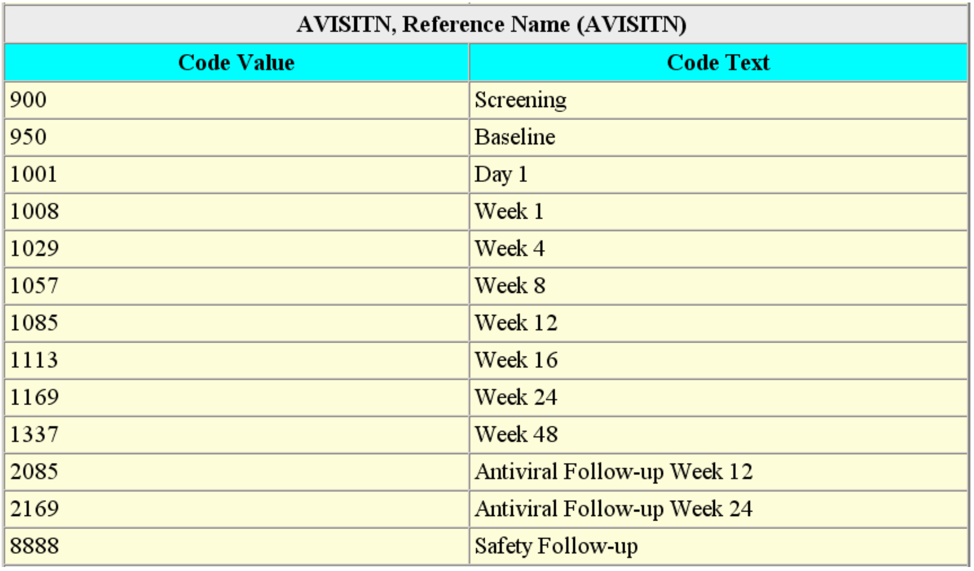
Generally, ADaM datasets have code-decode variable pairs, e.g., AETOXGRN and AETOXGR, where the former variable stores code values and the latter stores decoded values. Controlled terminology for the code-decode variable pairs will be defined by the sponsor with 1:1 mapping. Code variables in code-decode variable pairs are used as a sorting key for Tables, Figures, and Listings (TFLs) reporting purpose. The examples of the sponsor-defined controlled terminology are shown below from the Controlled Terminology Section of our define.xml. The code values are usually numeric or codes, which are used to decide the order of the decoded values shown in TFLs. Controlled term AETOXGRN in Display 4 defines 1:1 mapping of variables AETOXGR and AETOXGRN for reporting AE severity in the TFLs. AVISITN in Display 5 defines 1:1 mapping of variables AVISIT and AVISITN for reporting analysis visit windows in the ADaM BDS datasets.



ATOXGRN, Reference Name (ATOXGRN)

Code Value	Code Text
1	MILD
2	MODERATE
3	SEVERE
4	LIFE-THREATENING

Display 4. An Example of Sponsor-Defined Controlled Terminology for AETOXGRN



AVISITN, Reference Name (AVISITN)	
Code Value	**Code Text**
900	Screening
950	Baseline
1001	Day 1
1008	Week 1
1029	Week 4
1057	Week 8
1085	Week 12
1113	Week 16
1169	Week 24
1337	Week 48
2085	Antiviral Follow-up Week 12
2169	Antiviral Follow-up Week 24
8888	Safety Follow-up

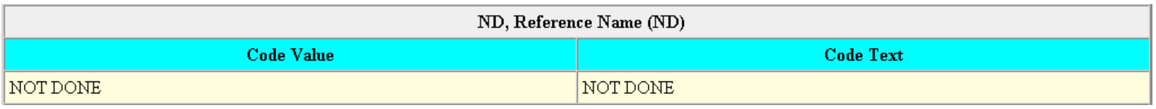
Display 5. An Example of Sponsor-Defined Controlled Terminology for AVISITN

3. Controlled Terminology Inherited from SDTM Domains

If the controlled terminology of a variable in an ADaM dataset is inherited from an SDTM domain and is not used in TFLs SAS programs, there is no need to create a corresponding code variable for it. It could be a CDISC/NCI code list, a sponsor defined code list in cases where standard vocabularies had not yet been defined, or an external code list.

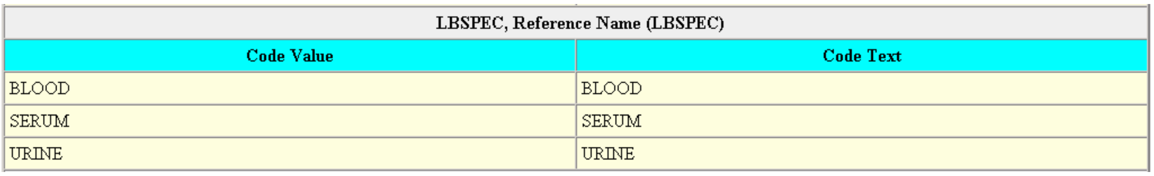
3.1. CDISC Codelist or Sponsor Defined Codelist Inherited from SDTM Domains

Display 6 shows an example of CDISC code list inherited from an SDTM Domain and Display 7 shows an example of sponsor-defined code list inherited from an SDTM Domain. The Code Values represent the values in the datasets, and they are usually identical to the Code Text in define.xml.



ND, Reference Name (ND)	
Code Value	**Code Text**
NOT DONE	NOT DONE

Display 6. An Example of CDISC Code List Inherited From an SDTM Domain for VSSTAT

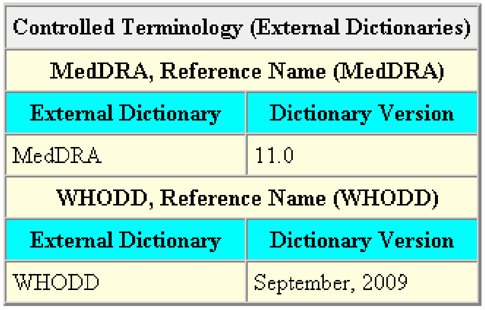


LBSPEC, Reference Name (LBSPEC)	
Code Value	**Code Text**
BLOOD	BLOOD
SERUM	SERUM
URINE	URINE

Display 7. An Example of Sponsor-Defined Controlled Terminology Inherited From an SDTM Domain for LBSPEC

3.2. External Code List - MedDRA and WHODD

The sponsor is expected to provide a subsection for external code list references in define.xml, like dictionary name and version, to be used to map the terms. Display 8 shows an example of the external published source, MedDRA and WHO dictionaries in define.xml.



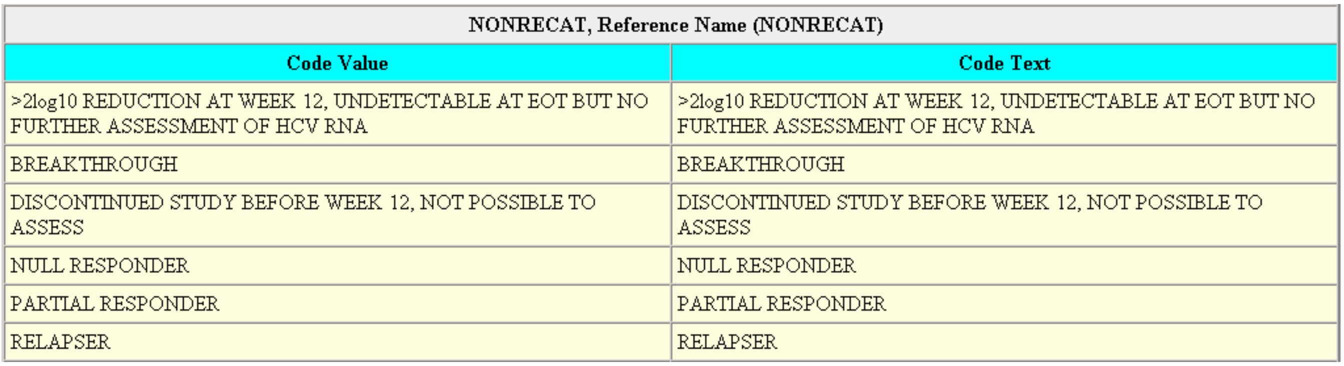
Controlled Terminology (External Dictionaries)	
MedDRA, Reference Name (MedDRA)	
External Dictionary	Dictionary Version
MedDRA	11.0
WHODD, Reference Name (WHODD)	
External Dictionary	Dictionary Version
WHODD	September, 2009

Display 8. An Example of External Code List Inherited From SDTM Domains

4. FDA defined therapeutic-specific Controlled Terminology

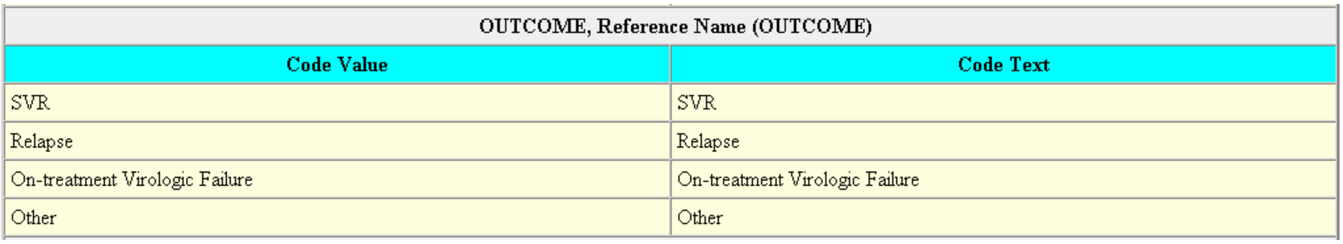
FDA defines therapeutic-specific controlled terminology to standardize the terms in a specific therapeutic area and further to facilitate the collaboration with the whole therapeutic area. These controlled terminologies are unique for ADaM datasets.

Display 9 shows an example of the controlled terminology given by FDA for Antiviral Information Management System (AIMS) datasets. It is for Non Responder Category of the study drug for Hepatitis C. Display 10 shows an example of the controlled terminology given by FDA for Drug Labeling. It is for the outcome category of the study drug for Hepatitis C. The Code Values are equal to the Code Text.



NONRECAT, Reference Name (NONRECAT)	
Code Value	Code Text
>2log10 REDUCTION AT WEEK 12, UNDETECTABLE AT EOT BUT NO FURTHER ASSESSMENT OF HCV RNA	>2log10 REDUCTION AT WEEK 12, UNDETECTABLE AT EOT BUT NO FURTHER ASSESSMENT OF HCV RNA
BREAKTHROUGH	BREAKTHROUGH
DISCONTINUED STUDY BEFORE WEEK 12, NOT POSSIBLE TO ASSESS	DISCONTINUED STUDY BEFORE WEEK 12, NOT POSSIBLE TO ASSESS
NULL RESPONDER	NULL RESPONDER
PARTIAL RESPONDER	PARTIAL RESPONDER
RELAPSER	RELAPSER

Display 9. An Example of FDA-Defined Therapeutic-Specific Controlled Terminology for NONRECAT

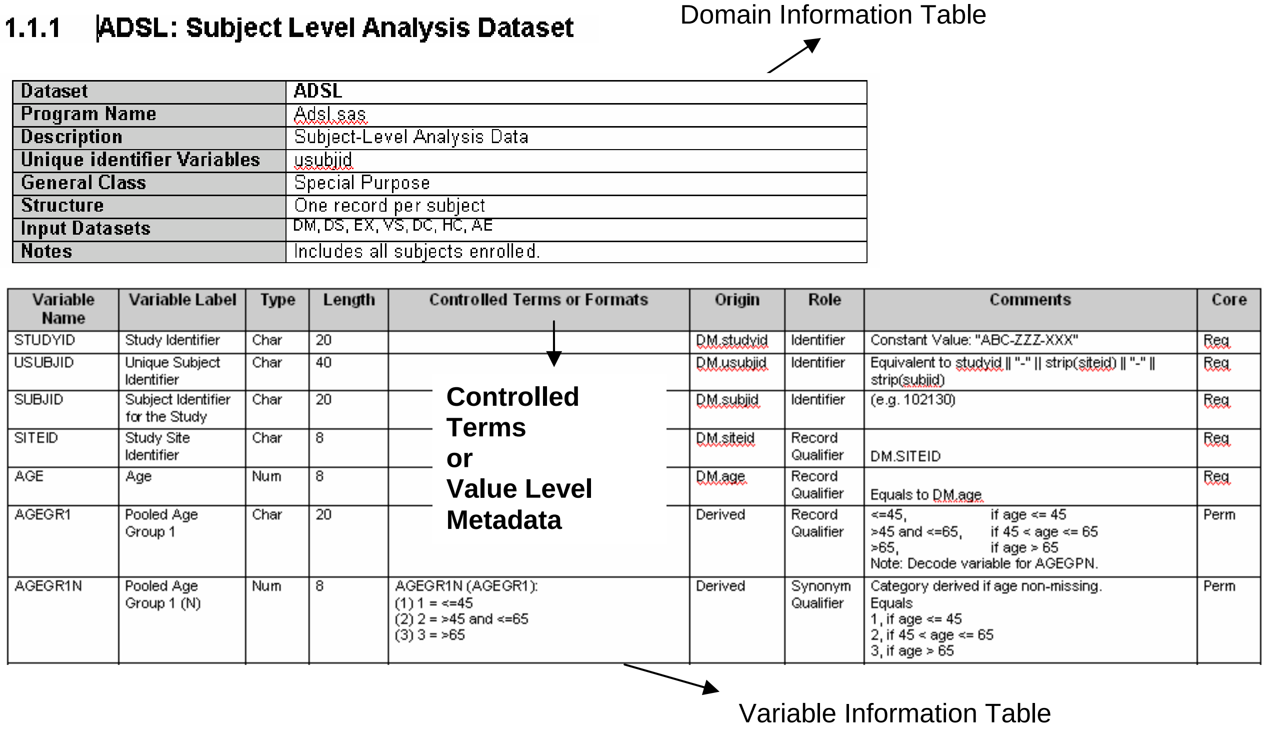


OUTCOME, Reference Name (OUTCOME)	
Code Value	Code Text
SVR	SVR
Relapse	Relapse
On-treatment Virologic Failure	On-treatment Virologic Failure
Other	Other

Display 10. An Example of FDA-Defined Therapeutic-Specific Controlled Terminology for OUTCOME

AN INTRODUCTION OF WORD® PROGRAMMING SPECIFICATION FOR ADAM

An individual programming specification for ADaM in MS Word® format facilitates programmers and statisticians to review and communicate derivation rules among them, as well as to track the changes. Display 11 shows the snapshot of an ADaM programming specification. The specification for each domain is composed of three parts: domain information table, variable information table, and an optional appendix or notes for a complex algorithm or derivation rules. Useful information in the first two parts will be used for ADaM automation purposes.


Domain Information Table
Controlled Terms or Value Level Metadata
Variable Information Table

1.1.1 ADSL: Subject Level Analysis Dataset

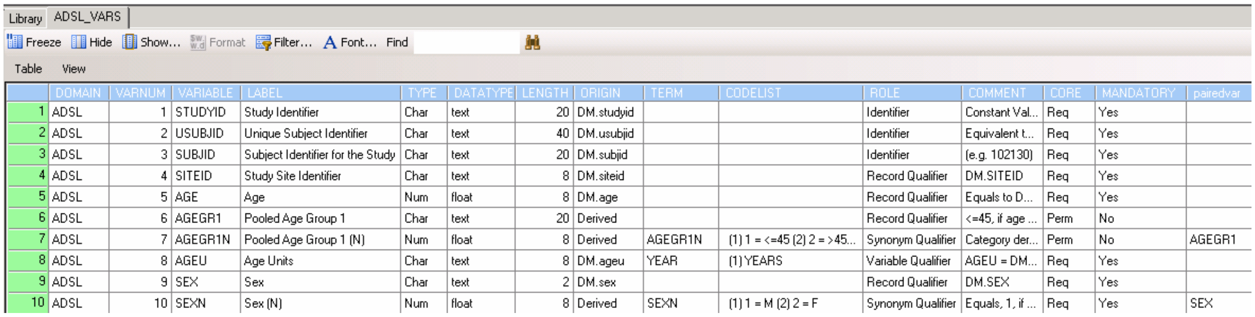
Dataset	**ADSL**
Program Name	Adsl.sas
Description	Subject-Level Analysis Data
Unique identifier Variables	usubjid
General Class	Special Purpose
Structure	One record per subject
Input Datasets	DM, DS, EX, VS, DC, HC, AE
Notes	Includes all subjects enrolled.

Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
STUDYID	Study Identifier	Char	20		DM.studyid	Identifier	Constant Value: "ABC-ZZZ-XXX"	Req
USUBJID	Unique Subject Identifier	Char	40		DM.usubjid	Identifier	Equivalent to studyid || "-" || strip(siteid) || "-" || strip(subjid)	Req
SUBJID	Subject Identifier for the Study	Char	20		DM.subjid	Identifier	(e.g. 102130)	Req
SITEID	Study Site Identifier	Char	8		DM.siteid	Record Qualifier	DM.SITEID	Req
AGE	Age	Num	8		DM.age	Record Qualifier	Equals to DM.age	Req
AGEGR1	Pooled Age Group 1	Char	20		Derived	Record Qualifier	<=45, if age <= 45 >45 and <=65, if 45 < age <= 65 >65, if age > 65 Note: Decode variable for AGEGPN.	Perm
AGEGR1N	Pooled Age Group 1 (N)	Num	8	AGEGR1N (AGEGR1): (1) 1 = <=45 (2) 2 = >45 and <=65 (3) 3 = >65	Derived	Synonym Qualifier	Category derived if age non-missing. Equals 1, if age <= 45 2, if 45 < age <= 65 3, if age > 65	Perm

Display 11. Individual Programming Specifications in Word® Format

In the domain information table, description of the domain will serve as the label of the ADaM dataset; in the variable information table, the variable name, label, type, and the length will define the variable attributes of the ADaM dataset. 'Controlled Terms or Formats' Column as the name implies specifies controlled terminologies for necessary variables and defines formats for date/time variables which will also be presented in define.xml.

The contents of the Word programming specification are converted into a SAS dataset named ADXX_VARS for automation process of our ADaM programming, as shown in Display 12.



	DOMAIN	VARNUM	VARIABLE	LABEL	TYPE	DATATYPE	LENGTH	ORIGIN	TERM	CODELIST	ROLE	COMMENT	CORE	MANDATORY	pairedvar
1	ADSL	1	STUDYID	Study Identifier	Char	text	20	DM.studyid			Identifier	Constant Val...	Req	Yes	
2	ADSL	2	USUBJID	Unique Subject Identifier	Char	text	40	DM.usubjid			Identifier	Equivalent t...	Req	Yes	
3	ADSL	3	SUBJID	Subject Identifier for the Study	Char	text	20	DM.subjid			Identifier	(e.g. 102130)	Req	Yes	
4	ADSL	4	SITEID	Study Site Identifier	Char	text	8	DM.siteid			Record Qualifier	DM.SITEID	Req	Yes	
5	ADSL	5	AGE	Age	Num	float	8	DM.age			Record Qualifier	Equals to D...	Req	Yes	
6	ADSL	6	AGEGR1	Pooled Age Group 1	Char	text	20	Derived			Record Qualifier	<=45, if age ...	Perm	No	
7	ADSL	7	AGEGR1N	Pooled Age Group 1 (N)	Num	float	8	Derived	AGEGR1N	(1) 1 = <=45 (2) 2 = >45...	Synonym Qualifier	Category der...	Perm	No	AGEGR1
8	ADSL	8	AGEU	Age Units	Char	text	8	DM.ageu	YEAR	(1) YEARS	Variable Qualifier	AGEU = DM...	Req	Yes	
9	ADSL	9	SEX	Sex	Char	text	2	DM.sex			Record Qualifier	DM.SEX	Req	Yes	
10	ADSL	10	SEXN	Sex (N)	Num	float	8	Derived	SEXN	(1) 1 = M (2) 2 = F	Synonym Qualifier	Equals, 1, if ...	Req	Yes	SEX

Display 12. Individual Programming Specification Converted to a SAS Dataset

HOW TO WRITE SPECIFICATION FOR CONTROLLED TERMINOLOGY IN ADAM PROGRAMING SPECIFICATION

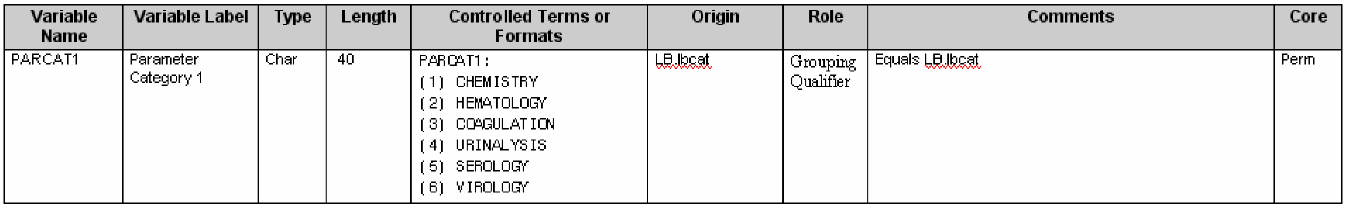
The 'Controlled Terms or Formats' Column specifies formats for date/time variables and the controlled terminologies. Formats must be ended with a trailing period '.', and the format of YYMMDD10. is for all the date variables, TIME5. is for all the time variables, and DATETIME20. is for all the date/time variables. The controlled terminology can be written by the following rules to help the SAS macro to automatically identify and retrieve the controlled terminologies defined in the specification.

1. Controlled Terminology for Value-Level Metadata for the ADaM Basic Data Structure (BDS) Datasets

The variable name (PARCAT1 and PARAMCD) is used as value list name in value level metadata for ADaM Basic Data Structure (BDS) Datasets. Hence writing "PARCAT1" or "PARAMCD" in the specification for the 'Controlled Terms or Formats' Column is optional as the SAS macro can handle the omission.

1.1 Value-Level Metadata for PARCAT1

Write "**PARCAT1:**" at the beginning, followed by the individual controlled terms (i.e., values) preceded by an ordering number '(#)'. As mentioned above inclusion of "PARCAT1" is optional . An example below shows how to fill in 'Controlled Terms or Formats' Column for **PARCAT1** in specification for ADLB.

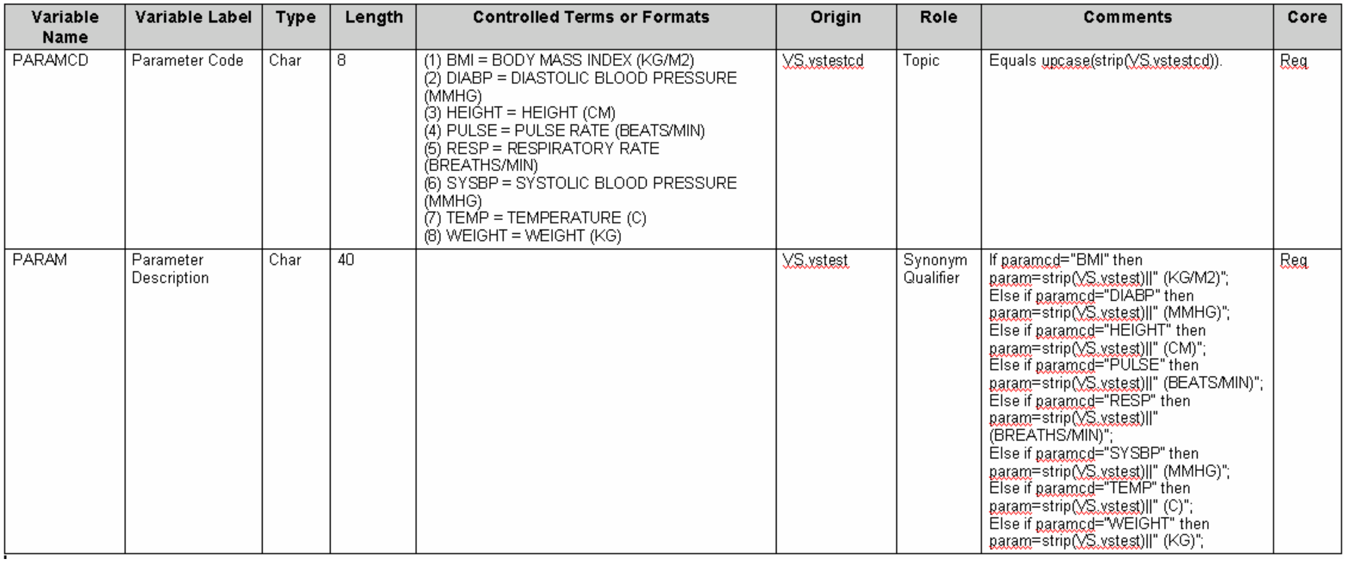


Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
PARCAT1	Parameter Category 1	Char	40	PARCAT1: (1) CHEMISTRY (2) HEMATOLOGY (3) COAGULATION (4) URINALYSIS (5) SEROLOGY (6) VIROLOGY	LB.lbcat	Grouping Qualifier	Equals LB.lbcat	Perm

Display 13. Illustration of PARCAT1 Controlled Terms in an ADaM Specification

1.2 Value-Level Metadata for PARAMCD

Write "**PARAMCD:**" at the beginning, use '=' to link the Value and Label from Value List of ADXX.PARAMCD. Inclusion of "PARAMCD" is optional. An example below shows how to fill in 'Controlled Terms or Formats' Column for **PARAMCD** and **PARAM** for ADVS in Display 3.



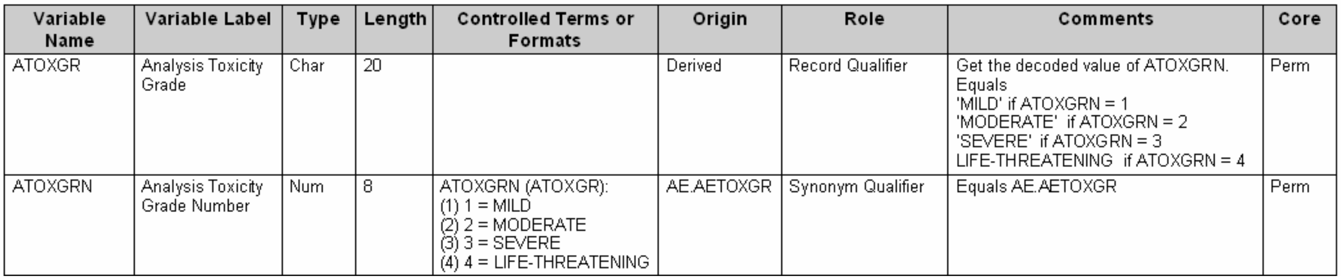
Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
PARAMCD	Parameter Code	Char	8	(1) BMI = BODY MASS INDEX (KG/M2) (2) DIABP = DIASTOLIC BLOOD PRESSURE (MMHG) (3) HEIGHT = HEIGHT (CM) (4) PULSE = PULSE RATE (BEATS/MIN) (5) RESP = RESPIRATORY RATE (BREATHS/MIN) (6) SYSBP = SYSTOLIC BLOOD PRESSURE (MMHG) (7) TEMP = TEMPERATURE (C) (8) WEIGHT = WEIGHT (KG)	VS.vstestcd	Topic	Equals upcase(strip(VS.vstestcd)).	Req

PARAM	Parameter Description	Char	40		VS.vstest	Synonym Qualifier	If paramcd="BMI" then param=strip(VS.vstest)||" (KG/M2)";
							Else if paramcd="DIABP" then param=strip(VS.vstest)||" (MMHG)";
							Else if paramcd="HEIGHT" then param=strip(VS.vstest)||" (CM)";
							Else if paramcd="PULSE" then param=strip(VS.vstest)||" (BEATS/MIN)";
							Else if paramcd="RESP" then param=strip(VS.vstest)||" (BREATHS/MIN)";
							Else if paramcd="SYSBP" then param=strip(VS.vstest)||" (MMHG)";
							Else if paramcd="TEMP" then param=strip(VS.vstest)||" (C)";
							Else if paramcd="WEIGHT" then param=strip(VS.vstest)||" (KG)";
								Req

Display 14. Illustration of PARAMCD Controlled Terms in an ADaM Specification

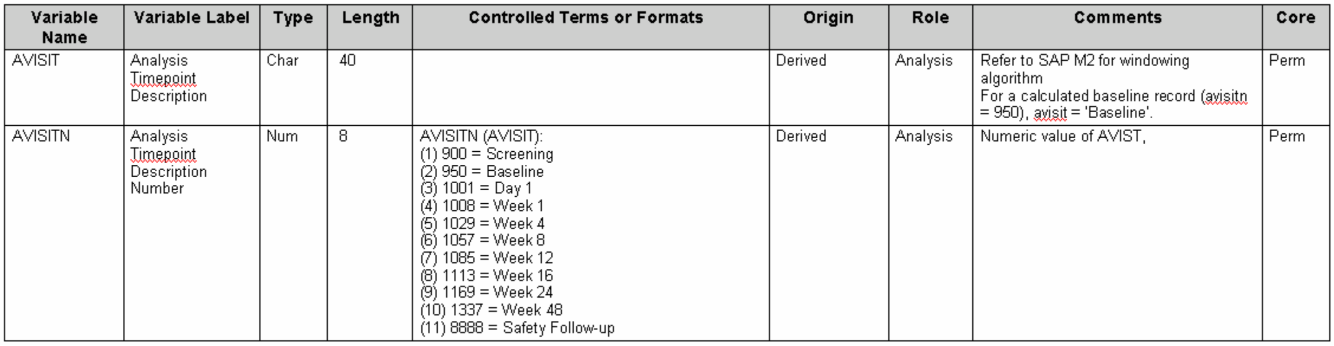
2. Sponsor-Defined Controlled Terminology

If the controlled terminology is defined for a pair of corresponding variables, fill in the column 'Controlled Terms or Formats' for the code variable only, leave it blank for the decoded variable, write "**code list name (decoded variable):**" at the beginning, followed by code value, '=' to link code value and code text, and code text preceded by an ordering number '(#)'. An example below shows how to fill in 'Controlled Terms or Formats' column for paired variables AETOXGR and AETOXGRN in Display 4. The code list name often uses the code variable name.



Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
ATOXGR	Analysis Toxicity Grade	Char	20		Derived	Record Qualifier	Get the decoded value of ATOXGRN. Equals 'MILD' if ATOXGRN = 1 'MODERATE' if ATOXGRN = 2 'SEVERE' if ATOXGRN = 3 LIFE-THREATENING if ATOXGRN = 4	Perm
ATOXGRN	Analysis Toxicity Grade Number	Num	8	ATOXGRN (ATOXGR): (1) 1 = MILD (2) 2 = MODERATE (3) 3 = SEVERE (4) 4 = LIFE-THREATENING	AE.AETOXGR	Synonym Qualifier	Equals AE.AETOXGR	Perm

Display 15. Illustration of Sponsor-Defined Controlled Terminology in an ADaM Specification Example 1



Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
AVISIT	Analysis Timepoint Description	Char	40		Derived	Analysis	Refer to SAP M2 for windowing algorithm For a calculated baseline record (avisitn = 950), avisit = 'Baseline'.	Perm
AVISITN	Analysis Timepoint Description Number	Num	8	AVISITN (AVISIT): (1) 900 = Screening (2) 950 = Baseline (3) 1001 = Day 1 (4) 1008 = Week 1 (5) 1029 = Week 4 (6) 1057 = Week 8 (7) 1085 = Week 12 (8) 1113 = Week 16 (9) 1169 = Week 24 (10) 1337 = Week 48 (11) 8888 = Safety Follow-up	Derived	Analysis	Numeric value of AVIST,	Perm

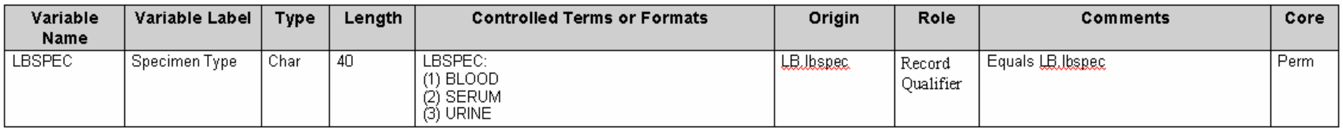
Display 16. Illustration of Sponsor-Defined Controlled Terminology in an ADaM Specification Example 2

3. Controlled Terminology Inherited from SDTM Domains

3.1. CDISC Codelist or Sponsor Defined Codelist Inherited from SDTM Domains

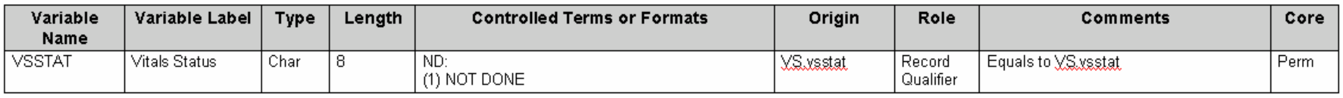
For a variable with controlled terminology inherited from SDTM Domains, provide the code list name with colon sign (:), followed by the individual controlled terms (i.e., code value) which is preceded by a number '(#)'. If no code list name is provided, then use the variable name for code list name.

The examples how to fill in 'Controlled Terms or Formats' Column for LBSPEC and VSSTAT are shown as follows:



Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
LBSPEC	Specimen Type	Char	40	LBSPEC: (1) BLOOD (2) SERUM (3) URINE	LB.lbspec	Record Qualifier	Equals LB.lbspec	Perm

Display 17. Illustration of Sponsor-Defined Codelist Inherited from CDISC SDTM Domain

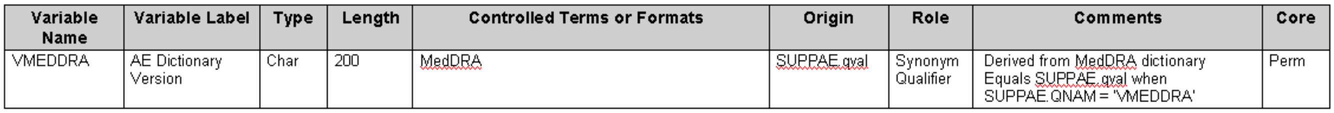


Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
VSSTAT	Vitals Status	Char	8	ND: (1) NOT DONE	VS.vsstat	Record Qualifier	Equals to VS.vsstat	Perm

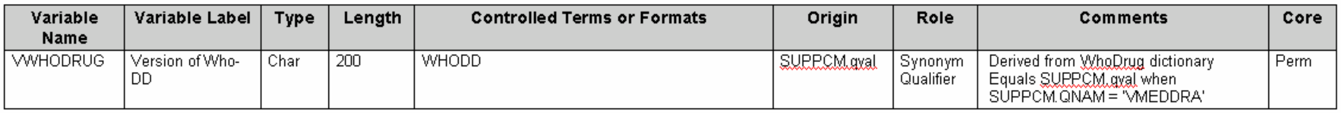
Display 18. Illustration of Controlled Terminology Inherited from CDISC SDTM Domain

3.2. External Code List - MedDRA and WHODD

Only the code list name is required for external code list, and the name of the external code list is case sensitive. The examples below are MedDRA and WHODD for external code list.



Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
VMEDDRA	AE Dictionary Version	Char	200	MedDRA	SUPPAE.qval	Synonym Qualifier	Derived from MedDRA dictionary Equals SUPPAE.qval when SUPPAE.QNAM = 'VMEDDRA'	Perm

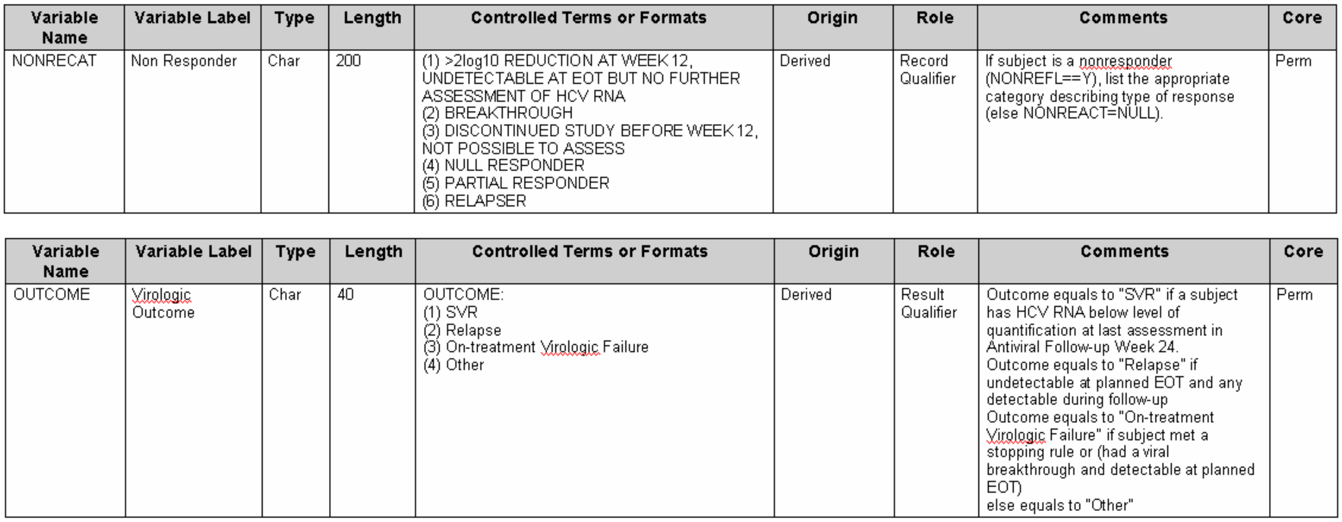


Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
VWHODRUG	Version of Who-DD	Char	200	WHODD	SUPPCM.qval	Synonym Qualifier	Derived from WhoDrug dictionary Equals SUPPCM.qval when SUPPCM.QNAM = 'VMEDDRA'	Perm

Display 19. Illustration of External Codelist in an ADaM Specification

4. FDA Defined Therapeutic-Specific Controlled Terminology

For FDA defined therapeutic-specific controlled terminology, provide the code list name with colon sign (:), followed by the individual controlled terms (i.e., code value) which is preceded by a number '(#)'. If no code list name is provided, then use the variable name for code list name. The examples below are for the controlled terminology: NONRECAT and OUTCOME.



Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
NONRECAT	Non Responder	Char	200	(1) >2log10 REDUCTION AT WEEK 12, UNDETECTABLE AT EOT BUT NO FURTHER ASSESSMENT OF HCV RNA (2) BREAKTHROUGH (3) DISCONTINUED STUDY BEFORE WEEK 12, NOT POSSIBLE TO ASSESS (4) NULL RESPONDER (5) PARTIAL RESPONDER (6) RELAPSER	Derived	Record Qualifier	If subject is a nonresponder (NONREFL==Y), list the appropriate category describing type of response (else NONREACT=NULL).	Perm

Variable Name	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Core
OUTCOME	Virologic Outcome	Char	40	OUTCOME: (1) SVR (2) Relapse (3) On-treatment Virologic Failure (4) Other	Derived	Result Qualifier	Outcome equals to "SVR" if a subject has HCV RNA below level of quantification at last assessment in Antiviral Follow-up Week 24. Outcome equals to "Relapse" if undetectable at planned EOT and any detectable during follow-up Outcome equals to "On-treatment Virologic Failure" if subject met a stopping rule or (had a viral breakthrough and detectable at planned EOT) else equals to "Other"	Perm

Display 20. Illustration of FDA Defined Therapeutic-Specifc Controlled Terminology

A MACRO TO RETRIEVE CONTROLLED TERMINOLOGY INFORMATION FROM THE ADAM PROGRAMMING SPECIFICATION

A macro %read_spec is developed to read the individual ADaM programming specification in CSV format, automatically retrieves domain information and variable information based on the standard structure of the given specification, and performs ADaM compliance checking with CDISC requirements and FDA submission requirements. The macro also retrieves and stores the controlled terminology information into a SAS variable level dataset, as shown in Display 12, which can be used for consistency checking purposes. A SAS dataset, named as ALL_VARS, will also be generated cumulatively each time when individual ADaM specification programs were run.

The macro call of %read_spec is as follows.

```
   %macro read_specs(indir=,specsnm=,outdir=,newdtnm=,runorder=);
```

Where

INDIR: Full Path for ADaM programming specification.
SPECSNM: Name of ADaM programming specification.
OUTDIR: Full Path for output SAS dataset which contains the attributes of ADaM variables.
NEWDTNM: A valid SAS dataset name for SAS dataset containing current specs information.
RUNORDER: A valid numeral, defining the order for a specific domain to run (in the final run).

For a code-decode variable pair, since decoded variables will share the same sponsor-defined controlled terminology with code variables, the macro %read_spec will retrieve the decoded variable name as well as the code list name from “Controlled Terms or Formats” Column. The snapshot of the code is shown below:

```
   data __temp;
      set specs;
      if index(term,'(') then do;
         pairedv = strip(scan(term,2,'()'));
         term = strip(scan(term,1,'()'));
      end;
   run;
```

A MACRO FOR AUTOMATIC CONSISTENCY CHECKING OF CONTROLLED TERMINOLOGY AND VALUE LEVEL METADATA BETWEEN ADAM DATASETS AND PROGRAMMING SPECIFICATION

A validation tool to check the proper use of Controlled Terminology and/or Value level metadata is developed to ensure the submission quality. It can be performed at any stage of the programming cycle in order to facilitate finalizing ADaM programming specifications earlier.

Macro **%ctlist_checking** compares the controlled terminology and the value level metadata defined in the ADaM Programming Specifications with ones in the ADaM datasets, detects any mismatches, and generates inconsistency report in RTF format if any exists.

The following is the macro call of a SAS macro for consistency checking of controlled terminology and value level metadata.

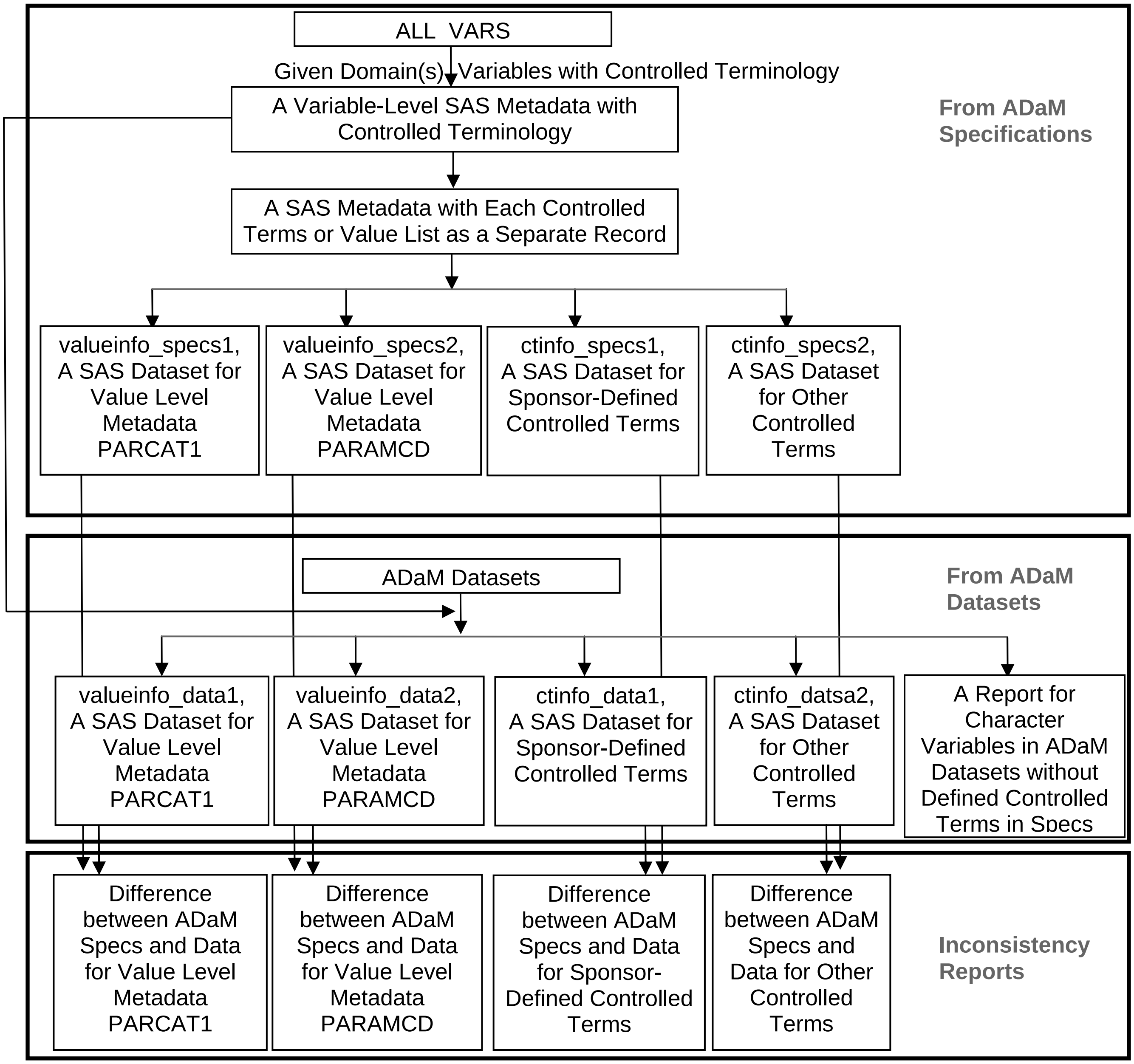
```
  %macro ctlist_checking(specdir = ,      /* a folder for programming specs.  */
                         datadir = ,      /* a folder for ADaM datasets       */
                         domain  = _ALL_  /* name of ADaM domain for checking */
                        );
```

Where,

SPECDIR: Full Path for ADaM Programming Specifications.
DATADIR: Full Path for ADaM datasets.
DOMAIN: An ADaM domain, If assigned _ALL_, all ADaM domains will be checked.

Consistency checking for controlled terminology can be performed during the development of individual ADaM dataset by assigning an ADaM dataset name to the macro variable &DOMAIN. If the macro variable &DOMAIN is not assigned a value, all ADaM domains will be checked for consistency of the controlled terminology and value level metadata, which is often but not necessarily done in the final run.

The programming flow chart is shown in Display 21.


ALL VARS
Given Domain(s) Variables with Controlled Terminology
A Variable-Level SAS Metadata with Controlled Terminology
From ADaM Specifications
A SAS Metadata with Each Controlled Terms or Value List as a Separate Record
valueinfo_specs1, A SAS Dataset for Value Level Metadata PARCAT1
valueinfo_specs2, A SAS Dataset for Value Level Metadata PARAMCD
ctinfo_specs1, A SAS Dataset for Sponsor-Defined Controlled Terms
ctinfo_specs2, A SAS Dataset for Other Controlled Terms
ADaM Datasets
From ADaM Datasets
valueinfo_data1, A SAS Dataset for Value Level Metadata PARCAT1
valueinfo_data2, A SAS Dataset for Value Level Metadata PARAMCD
ctinfo_data1, A SAS Dataset for Sponsor-Defined Controlled Terms
ctinfo_datsa2, A SAS Dataset for Other Controlled Terms
A Report for Character Variables in ADaM Datasets without Defined Controlled Terms in Specs
Difference between ADaM Specs and Data for Value Level Metadata PARCAT1
Difference between ADaM Specs and Data for Value Level Metadata PARAMCD
Difference between ADaM Specs and Data for Sponsor-Defined Controlled Terms
Difference between ADaM Specs and Data for Other Controlled Terms
Inconsistency Reports

Display 21. Programming Flow Chart for Macro %ctlist_checking

There are some notes about the logics behind the %ctlist_checking macro.

1. The comparison of controlled terminology will ONLY be performed for the variables with ‘Controlled Terms or Formats’ Column filled in. Therefore, the macro reads a SAS dataset, ALL_VARS, which is cumulatively generated to incorporate all the variable information from all existed domains, and selects the target domain and target variables for consistency checking.

```
data ctinfo_specs valueinfo_specs;
     set speclib.all_vars;
     where term ne '' and substr(reverse(strip(term)),1,1) ne '.' and term not in
           ('MedDRA','WHODD') %if %upcase(&domain.) ne _ALL_ %then and
           domain=upcase("&domain.");;
     if variable in ('PARAMCD','PARCAT1') then output valueinfo_specs;
     else output ctinfo_specs;
run;
```

2. Each controlled terms or value lists from ADaM specifications will be retrieved as a separate record.

3. Retrieve the code lists from the ADaM datasets ONLY if they are defined in the programming specifications. The following code retrieves the ADaM datasets and variables for which the controlled terminology is defined.

```
*** get the codelists from dataset ***;
proc contents data=datalib.&domain. noprint out=data_allvars;run;
data data_allvars;
     length domain variable $8. label $40.;
     set data_allvars(rename=(label=var_label));
     domain=strip(memname); variable=strip(name); label=strip(var_label);
run;
proc sort data=data_allvars; by domain variable; run;
proc sort data = ctinfo_specs out = specs_ctvars(keep=domain variable) nodupkey;
     by domain variable;
run;
/* Retrieve controlled terminology from ADaM datasets */
data data_ctvars;
     merge data_allvars(in=a) specs_ctvars(in=b); by domain variable; if a and b;
run;
```

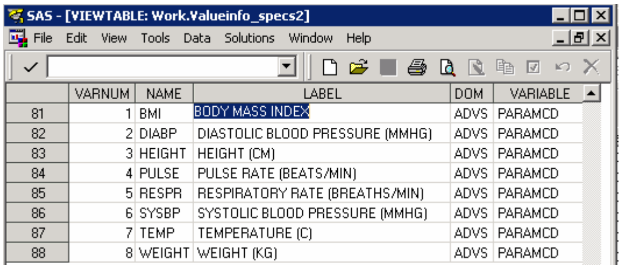
4. Retrieve the value lists from the ADaM datasets for PARCAT1 and PARAMCD, respectively.

The following code retrieves the ADaM datasets and variables which contain the value level metadata for PARCAT1 and PARAMCD.

```
*** For value level list: PARAMCD ***;
proc sort data= data_allvars(where=(variable in
('PARCAT1','PARAMCD')))out=value_dsnames;
     by domain;
run;
```

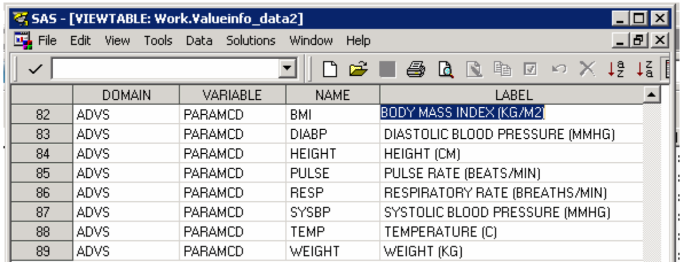
5. Compare the code lists from the ADaM datasets with ones from the programming specifications, detect the mismatches, and output non-consistency reports for both controlled terminology and the value level metadata.

Display 22 and Display 23 shows two intermediate SAS datasets for Value Level Metadata PARAMCD from ADaM specifications and datasets, respectively, for ADVS dataset.



	VARNUM	NAME	LABEL	DOM	VARIABLE
81	1	BMI	BODY MASS INDEX	ADVS	PARAMCD
82	2	DIABP	DIASTOLIC BLOOD PRESSURE (MMHG)	ADVS	PARAMCD
83	3	HEIGHT	HEIGHT (CM)	ADVS	PARAMCD
84	4	PULSE	PULSE RATE (BEATS/MIN)	ADVS	PARAMCD
85	5	RESPR	RESPIRATORY RATE (BREATHS/MIN)	ADVS	PARAMCD
86	6	SYSBP	SYSTOLIC BLOOD PRESSURE (MMHG)	ADVS	PARAMCD
87	7	TEMP	TEMPERATURE (C)	ADVS	PARAMCD
88	8	WEIGHT	WEIGHT (KG)	ADVS	PARAMCD

Display 22. Value Level Metadata PARAMCD from ADVS Specification


SAS - [VIEWTABLE: Work.Valueinfo_data2]
File Edit View Tools Data Solutions Window Help

	DOMAIN	VARIABLE	NAME	LABEL
82	ADVS	PARAMCD	BMI	BODY MASS INDEX (KG/M2)
83	ADVS	PARAMCD	DIABP	DIASTOLIC BLOOD PRESSURE (MMHG)
84	ADVS	PARAMCD	HEIGHT	HEIGHT (CM)
85	ADVS	PARAMCD	PULSE	PULSE RATE (BEATS/MIN)
86	ADVS	PARAMCD	RESP	RESPIRATORY RATE (BREATHS/MIN)
87	ADVS	PARAMCD	SYSBP	SYSTOLIC BLOOD PRESSURE (MMHG)
88	ADVS	PARAMCD	TEMP	TEMPERATURE (C)
89	ADVS	PARAMCD	WEIGHT	WEIGHT (KG)

Display 23. Value Level Metadata PARAMCD from ADVS Dataset

Displays 24 - 27 show a typical report of non-consistency between AdaM datasets and specifications. Decision will be made to update either the programming specifications or the ADaM derivation program to handle these mismatches, which will be explained in the next section.

The following PARCAT1 Variables with Different Value Level Metadata between Programming Specs. and Datasets

Domain	Variable	Value	Value Label in Dataset	Value Label in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.
ADLB	PARCAT1	VIROLOGY	VIROLOGY			Yes

Display 24. Non-Consistency Report of Value List Metadata for PARCAT1 Between ADaM Datasets and Specifications

The following PARAM Variables with Different Value Level Metadata between Programming Specs. and Datasets

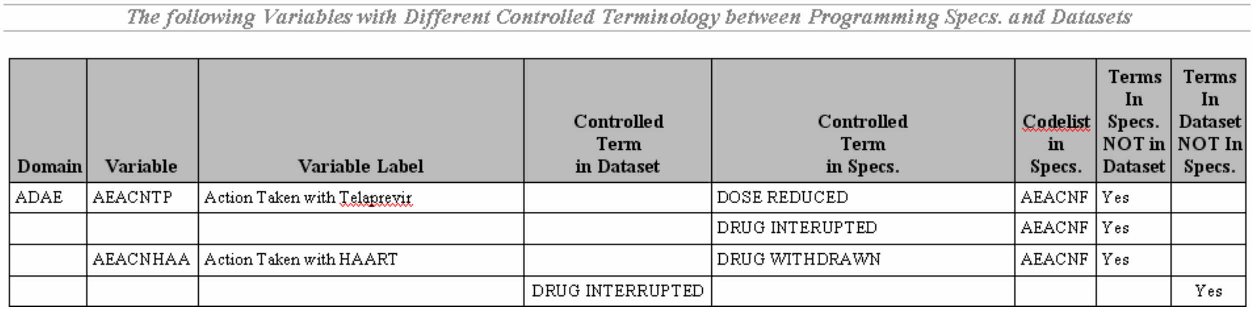
Domain	Variable	Variable Label	Value	Value Label in Dataset	Value Label in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.	Different Controlled Terminology
ADVS	PARAMCD	Parameter Code	RESP	RESPIRATORY RATE (BREATHS/MIN)			Yes	
			BMI	BODY MASS INDEX (KG/M2)	BODY MASS INDEX			Yes
			RESPR		RESPIRATORY RATE (BREATHS/MIN)	Yes		

Display 25. Non-Consistency Report of Value List Metadata for PARAMCD Between ADaM Datasets and Specifications

The following Coded Variables with Different Decoded Terminology between Programming Specs. and Datasets

Domain	Variable	Variable Label	Coded Controlled Term	Decoded Controlled Term in Dataset	Decoded Controlled Term in Specs.	Codelist in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.	Different Decoded Terminology
ADAE	AEACNN	Action with Study Treatment Number	5	NOT APPLICABLE				Yes	
	AEOUTN	Outcome of Adverse Event Number	3	RECOVERED/ RESOLVED WITH SEQUELAE	RECOVERED/RESOLVED WITH SEQUELAE	AEOUTN			Yes
			4		FATAL	AEOUTN	Yes		
			5		UNKNOWN	AEOUTN	Yes		
ADEG	BQTGR1N	Baseline Pooled QT Group 1 (N)	4	> 500	> 500 msec	BQTGR1N			Yes
ADSL	OUTCOMEN	Virologic Outcome (N)	3		Rebound at EOT+12	OUTCOMEN	Yes		
			4		Rebound at EOT+24	OUTCOMEN	Yes		

Display 26. Non-Consistency Report of Sponsor Defined Controlled Terminology Between ADaM Datasets and Specifications



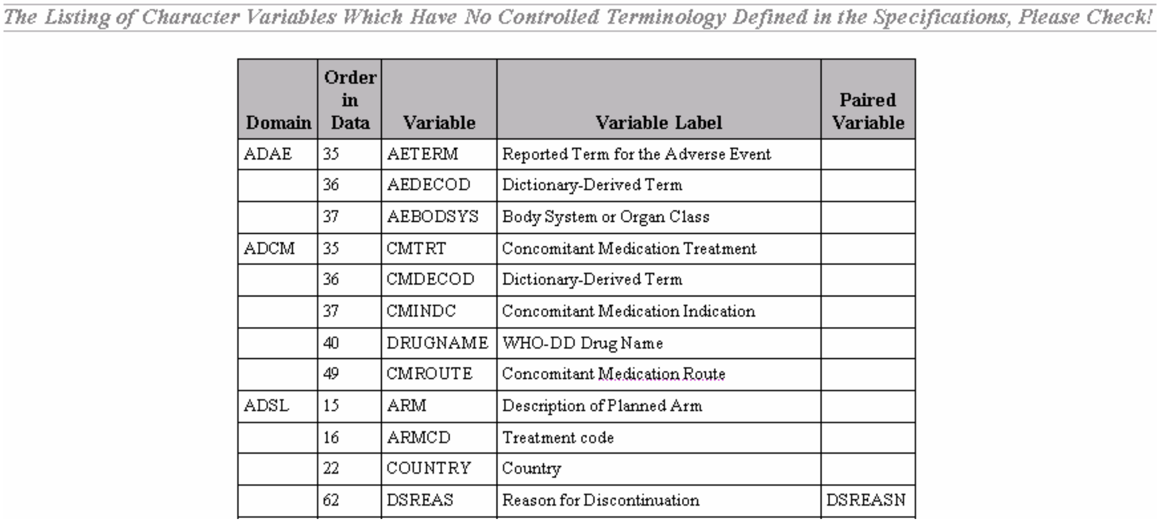
The following Variables with Different Controlled Terminology between Programming Specs. and Datasets

Domain	Variable	Variable Label	Controlled Term in Dataset	Controlled Term in Specs.	Codelist in Specs.	Terms In Specs. NOT in Dataset	Terms In Dataset NOT In Specs.
ADAE	AEACNTP	Action Taken with Telaprevir		DOSE REDUCED	AEACNF	Yes	
				DRUG INTERUPTED	AEACNF	Yes	
	AEACNHAA	Action Taken with HAART		DRUG WITHDRAWN	AEACNF	Yes	
			DRUG INTERRUPTED				Yes

Display 27. Non-Consistency Report of CDISC or FDA defined Controlled Terminology Between ADaM Datasets and Specifications

6. Report any character variables without defining controlled terminology or value level metadata for manual review to identify the omissions of specification for controlled terminology or value level metadata in programming specifications. Variables USUBJID, SUBJID, SITEID, and the variables end with DTC should be excluded in the report. Core variables in ADaM datasets other than ADSL should be excluded, too, for their attributes and controlled terminology have been checked in ADSL. The programmers should review the report and check these variables to make sure whether they should have the Controlled Terms or Format Column filled or not. Once these are identified in the reviewing, the corresponding controlled terminology should be written in specifications.

Display 28 shows a typical report for character variables without specification for controlled terminology in specifications. Controlled terminology for CMROUTE, ARMCD and ARM, and DSREAN and DSREAS should be added in the “Controlled Terms or Formats” Column in the programming specifications.



The Listing of Character Variables Which Have No Controlled Terminology Defined in the Specifications, Please Check!

Domain	Order in Data	Variable	Variable Label	Paired Variable
ADAE	35	AETERM	Reported Term for the Adverse Event	
	36	AEDECOD	Dictionary-Derived Term	
	37	AEBODSYS	Body System or Organ Class	
ADCM	35	CMTRT	Concomitant Medication Treatment	
	36	CMDECOD	Dictionary-Derived Term	
	37	CMINDC	Concomitant Medication Indication	
	40	DRUGNAME	WHO-DD Drug Name	
	49	CMROUTE	Concomitant Medication Route	
ADSL	15	ARM	Description of Planned Arm	
	16	ARMCD	Treatment code	
	22	COUNTRY	Country	
	62	DSREAS	Reason for Discontinuation	DSREASN

Display 28. Report of Character Variables without Controlled Terminology in ADaM Specifications

DECISION MAKING ON THE MISMATCHES BETWEEN ADAM SPECS AND DATASETS

There are 5 scenarios of mismatches between ADaM datasets and specifications.

1. The Controlled Terms or Value Lists are not in the Datasets but in the Specifications.

Usually, all values in the permissible value set for the study should be included, whether they are represented in the submitted data or not. Therefore, those code lists correctly defined in the programming specifications but not shown in the ADaM datasets are acceptable, and no further action is needed for them. The examples can be found in Display 27 for ADAE.AEACNTP and Display 26 for ADSL.OUTCOMEN.

2. The Controlled Terms or Value Lists are in the Datasets but not in the Specifications.

The specification does not list all the possible values for the controlled terms or value lists. This kind of mismatches is not acceptable, and adding the missing controlled terms or value lists in the specifications is the solution. An example is shown in Display 24 for ADLB.PARCAT1.

3. The Code Value for Sponsor-Defined Controlled Terminology or the Value for Value Level Metadata PARAMCD are Differently Defined in the Datasets from that in the Specifications.

This kind of mismatches is not acceptable, and the revision should be done in either the specifications or the datasets to make them consistent. An example can be found in Display 25 for RESPIRATORY RATE from ADVS.PARAMCD. PARAMCD uses “RESP” in the dataset vs. “RESPR” in the specifications. To change “RESPR” in the specifications to “RESP” resolves the mismatch.

4. The Decoded Value for Sponsor-Defined Controlled Terminology or the Value Label for Value Level Metadata PARAMCD are Differently Defined in the Datasets from that in the Specifications.

This kind of mismatches is not acceptable, and the revision should be done in either the specifications or the datasets to make them consistent. An example can be found in Display 26 for code value 3 from ADAE.AEOUTN. The ADaM program ADAE.SAS need to be updated to revise the decoded value ‘ RECOVERED/RESOLVED WITH SEQUELAE’ to ‘RECOVERED/ RESOLVED WITH SEQUELAE’

5. Typo Occurrence either in ADaM Specifications or in ADaM Derivation Programs

Correct the typos. An example can be shown in Display 27 for controlled terms DRUG INTERRUPTED from ADAE.AEACNTP. The typo “INTERUPTED“ in specification should be corrected to “INTERRUPTED”.

A summary of these 5 scenarios is shown in Table 1.

#	Scenario	Condition	Action Taken
1	Controlled Terms or Value Lists are not in the Datasets but in the Specifications	Code lists are correctly defined in specifications	No Action Needed
2	Controlled Terms or Value Lists are in the Datasets but not in the Specifications	Specification does not list all the possible values for the controlled terms or value lists	Add Missing Controlled Terms or Value Lists to Specifications
3	Code Value for Sponsor-Defined Controlled Terminology or Value for Value Level Metadata PARAMCD are Differently Defined in the Datasets from that in the Specifications	Code Value or Value in datasets is not consistent with Standard Controlled Terms	Revise ADaM Datasets
		Code Value or Value in specifications is not consistent with Standard Controlled Terms	Revise ADaM Specifications
4	Decoded Value for Sponsor-Defined Controlled Terminology or Value Label for Value Level Metadata PARAMCD are Differently Defined in the Datasets from that in the Specifications	Decode Value or Value Label in datasets is not consistent with Standard Controlled Terms	Revise ADaM Datasets
		Decode Value or Value Label in specifications is not consistent with Standard Controlled Terms	Revise ADaM Specifications
5	Typo Occurs Either in ADaM Specifications or in ADaM Derivation Programs		Correct the typo

Table 1. Summary of 5 Scenarios of Mismatches between ADaM Datasets and Specifications

CONCLUSION

In summary, this paper classifies controlled terminology in ADaM datasets into four categories, and introduces how to write ADaM programming specifications for controlled terminology and a SAS macro for automatic consistency checking of them between ADaM datasets and programming specification, and further between ADaM datasets and define.xml. It also provides innovative solutions for mismatches the macro detects.

This macro-based comprehensive approach can ensure consistency between ADaM datasets and define.xml for final FDA submission. Since it can be used at any stage of the programming cycle, the high quality of the submissions can be achieved in a cost-effective and efficient way. We hope this approach can assist you in handling ADaM controlled terminology and value level metadata in order to enhance the submission quality.

REFERENCES

“CDISC SDTM/ADaM Pilot Project, Project Report”- http://www.cdisc.org/stuff/contentmgr/files/0/df91a087c6df43275288267c9fe92180/misc/sdtmadampilotprojectreport.pdf

CDISC Analysis Data Model Team. “Analysis Data Model (ADaM) Implementation Guide”. December 2009. http://www.cdisc.org/adam

Xiangchen (BoB) Cui, Min Chen, and Tathabbai Pakalapati. “An Innovative ADaM Programming Tool for FDA Submission”, PharmaSUG, May 2012

Min Chen, Xiangchen Cui, Scott Moseley. (2011) "Automating the Process of Preparing Data Definition Document for NDA Electronic Submission from Programming Specification in Word Format", PharmaSUG, May 2011.

John R. Gerlach. (2011) "Validating Controlled Terminology in SDTM Domains", PharmaSUG, May 2011.

ACKNOWLEDGEMENTS

Appreciation goes to Kelly Blackburn, Stacy Surensky, Anna Legedza and Tathabbai Pakalapati for their review and comments.

CONTACT INFORMATION

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

Name: Xiangchen (Bob) Cui**,** Ph.D.
Enterprise: Vertex Pharmaceuticals, Inc.
Address: 88 Sidney Street
City, State ZIP: Cambridge MA, 02139
Work Phone: 617-444-6069
Fax: 617-460-8060
E-mail:xiangchen_cui@vrtx.com

Name: Min Chen**,** Ph.D.
Enterprise: Vertex Pharmaceuticals, Inc.
Address: 88 Sidney Street
City, State ZIP: Cambridge MA, 02139
Work Phone: 617-444-7134
Fax: 617-460-8060
E-mail: min_chen@vrtx.com