



Development Solutions

The Use of Metadata in Creating, Transforming and Transporting Clinical Data

Gregory Steffens
Director, Global SAS Programming
ICON Development Solutions



Development Solutions

Data Flow Constituents

- Data flow is comprised of transformation and derivation
 - Transformation – restructures, re-attributes and integrates
 - Derivation – creates new values in a target database that do not exist in the source



Development Solutions

Data Transformations

- Data transformations comprise a large portion of what we do
 - Extract from data collection/cleaning applications to a standard (e.g. SDTM)
 - Integrate ancillary data like lab, randomization, QOL, etc.
 - Creation of the analysis database (e.g. ADaM)
 - Integrate multiple studies into an integrated database



Development Solutions

The DTE

- Therefore, there is a need to define, create, validate and describe databases efficiently and with a consistently high level of quality
- A combination of reusable code driven by standardized metadata is the answer – the Data Transformation Engine (DTE)
- The code must not make any assumptions of the source and target databases, this is communicated to the code via metadata
- The metadata structures must be absolutely standardized across all applications and data standards
- Metadata is populated prescriptively and drives the data flow. Data specifications are entered once and used many times.



Development Solutions

Metadata Constituents

- A standard list of attributes to include in any description of a database or of a data standard
- Put in a standard set of data structures that can be read by programming code
- The attributes must be highly structured in order to be usable by DTE code
- To define a standard for defining data standards
- Enables easy publication in different formats, html, word, pdf, xml, etc. Generate documents from metadata, not metadata from documents!



Development Solutions

Standard Database Attributes

- Data Set Level**
 - Short/long names, data set location, order
- Variable level**
 - Short/long name, type, length label, primary key flag, format, value list name, suppqal flag, code/decode relationship, order, aCRF location, etc.
- Valid values**
 - Value list name, start/end value, short decode, long decode, rank
- Descriptions**
 - Source name, derivation description
- Row-level attributes**
 - Identical to variable level attributes but for subsets of rows defined by a parameter variable value

Row-Level Metadata



Development Solutions

- Necessary to fully describe tall-thin data set structures

USUBJID	SYSBP	BPSYSLOC	BPSYSU	HEIGHT	HEIGHTU	WEIGHT	WEIGHTU	BMI	BMIU
1	120	STANDING	mm Hg	185	CM	90	KG	26.3	Kg/m ²

USUBJID	VSTESTCD	VSLOC	VSORRES	VSORRESU
1	SYSBP	STANDING	120	mm Hg
1	HEIGHT		185	CM
1	WEIGHT		90	KG
1	BMI		26.3	Kg/m ²

Metadata Structure



Development Solutions

- Structured content to enable programmatic access
- Storage structure is separate from publication structure – maximize programmatic access and user friendly access by people
- Also separate from entry format
- Maximize sharing of information within the metadata, e.g. values lists and descriptions. Columns to the right are not a good design for shared information.

Some Principles of Metadata Design



Development Solutions

- Rigorously standardized for all database and standard descriptions, no metadata design change for database types!
- Maximize structured information and programmatic access, e.g. primary keys flagged instead of listed
- Enter once; use many. e.g. descriptions and values
- Derivation logic in descriptions though

Objectives of Metadata



Development Solutions

- It is critical to explicitly define the objectives. Many disagreements arise from an unstated difference in assumed objectives
- Objectives allow evaluation of the success of the metadata design; e.g. retrospective description for resubmission or prescriptive enabler of automation.

Objectives ...



Development Solutions

- Single metadata design for everything
- Minimize duplicate entry
- Include enough attributes to enable automation
- Separate metadata design from publication and from entry
- Generalize SAS macros, don't specialize for any specific database standard. Automate with SAS macro code that does not make assumptions about the data it is transforming. Don't automate each domain, automate all domains.

Steps in the Metadata Process Getting approval



Development Solutions

- Where to start study metadata?
 - Get metadata from core standard, a compound standard, a similar study's metadata with %mdmake or use %mdbuild and %mdfreqvals on a similar study that does not have metadata
- Publish data specification with %mdprint for review ...revise ... republish ... approval
- Call %mdcompare to compare your study to other studies and standards
- Critical to store all the data standards in one metadata structure so metadata can be copied and compared
- Use a GUI to maintain the metadata- enter, change, delete and copy

Steps-Build the ADS Data Sets



Development Solutions

- Call %mdmkdsn to create 0-observation data sets
- Call %mdatribs to create all the labels, format associations, types, lengths, sort by keys, order variables in PDV, etc.
- Call %mdformats to create a format catalog
- Call %dt_make_decodes to create decode variables
- Dt_copy_headers to copy header variables
- (Ut_saslogcheck to evaluate SAS log files)

Steps-Validate the ADS Data Sets



Development Solutions

- Call %mdcheck to compare the ADS data sets to the specification contained in metadata
 - This calls %mdbuild, %mdcompare and then does a comparison of values
 - Compares the attributes defined in the metadata to the attributes that exist in the described data sets. Many attributes, including row-level comparisons and valid values.

Steps-Regulatory and Outside Standards



Development Solutions

- The data information is required by the FDA electronic submissions (define.pdf or define.xml)
- We have to do it anyway so why not get the most value?
 - The study (and ISS) metadata are used internally to build and validate the database
 - Quality is better and more value from the work
 - Can generate CDISC, define.xml from the metadata

Some Automation that is Enabled



Development Solutions

- Creating study data requirements from data standards
- Publish data requirements in several formats – pdf, xml, html, etc.
- Publish data requirements with different levels of detail for different user types
- Implement all database attributes from the requirements with one macro call
- Create version 5 transport files with renames and length changes of names and labels
- Create integrated databases
- Create all decode variables with one macro call

More Automation Examples



Development Solutions

- Move variables in and out of suppqul domains
- Validate that a database follows a standard and a study requirement
- Compare a new data requirement to a standard or other studies
- Create 0-obs data sets
- Automate TFL shells
- Transform databases from any structure to any other structure ... but what else do we need to do that?

Map Metadata The Next Kind of Metadata



Development Solutions

- So far, we have metadata that describes databases
- The next step is standard map metadata to define transformation types from one database to another
- Map metadata associates one observation of source metadata with one or more observations of target metadata. Simple structure!
- This mapping information is read by SAS macros that generate all the SAS code to create a target database from a source database
- Map metadata should be separate but integrated with metadata.

The DTE Code Examples



Development Solutions

- %dtmap
 - (source_mdlib=m,source_prefix=raw_,target_mdlib=m,target_prefix=target_,maplib=me,inlib=raw,outlib=sdm,supqual_make=yes)
- %mdprint
 - (mdlib=m,inprefix=target_,html=yes,htmlfile=c:\metadata\target_defne.html)
- %md2odm
 - (mdlib=m,outxml=c:\metadata\SDTM.xml,StudyName=ICR_STDM_Standard 3.1.2,ProtocolName=SDTM,StandardName=SDTM 3.1.2,StudyDescription=SDTM Standard,defineVersion=1.0.0,odmversion=1.2,crt_prefix=def)

Advantages of the DTE



Development Solutions

- Standards are a means to automation and automation is a means to efficiency and quality
- Much less code invented and validated for individual studies
- Change in skill set requirements for data flow – separate coding skills from database and clinical skills
- More productivity from junior staff
- Faster, better and cheaper.

Metadata and Map Metadata



Development Solutions

- Metadata defines source and target databases
- Map metadata defines connections between metadatabases for the automation of database transformations
 - Change values of variables and paramrels
 - Change values of parameter variables
 - Rename variables and data sets
 - Move variables between same-keyed data sets
 - Rekey variables!
 - Re-attribute variables and data sets
 - Keep variables
 - Drop variables

Types of Re-Keying



Development Solutions

- Target data set has the same key variables as the source data set (even if renamed)
- Target data set has a superset of the source data set primary keys
- Target data set has a subset of the source data set primary keys
- Target data set has a subset of the source data set primary keys and other primary keys besides
- Target data set has no primary keys in common with the source data set
- Target data set has a foreign key that is a primary key in the source data set
- Keep in mind that the primary keys might be renamed in the target data set

Examples of macros



Development Solutions

List of some of the macros and their functionality which help us to achieve efficiency and ensure good quality:

Mdprint/md2odm	Publish in html or xml format
Mdatrbs	Apply attributes defined in metadata to a data library
Ut_find_decodes	Finds decode variables and their attributes
Dt_make_decodes	Creates decode variables
Dt_copy_headers	Copies header variables from source to target data sets
Mdcompare / mdcompare_print	Compares metadatabases to each other, such as a study requirement to a standard or a study to a study
mdcheck	Checks data and reports discrepancies with the metadata
mdbuild	Builds metadata to describe an existing data library
mdfreqvals	Creates the values metadata set (supplements mdbuild)

Examples of macros



Development Solutions

dtmap	Top level macro that users call to transform data from one format to another, e.g. raw to SDTM to ADaM to IDB
Dt_thin2wide	Convert tall-thin to short-wide
Dt_wide2thin	Convert short-wide to tall-thin
Tool_code_lib	Documents program code
Ut_saslogcheck	Checks SAS logs for disallowed messages
Ut_age_years	Computes age in years
Ut_truncate_long_chars	Truncates long character variable lengths to least length to hold longest value
mdport	Creates a transport file of a metadatabase to archive versions
md2excel / excel2md	Converts metadata between SAS and excel
mdmkdsn	Creates 0-observation data sets as defined in metadata

Examples of macros	
	
Development Solutions	
Suppqual_make	Creates the suppqual data sets, by reading the suppqual flag in the metadata to identify supplementary qualifiers
Suppqual_get	Gets supplementary qualifier variables from the suppqual data sets and adds them to their proper domain
Dtmap_values	Changes the value of variables by reading value map metadata
Mdformats	Create user formats from values metadata set
Missvars	Report variables that have a missing value in all observations
Missobs	Report observations where all variables have a missing value