



Implementing CDISC Based Processes:  
General Solutions for Common Challenges

## **From Source Data to SDTM**

Steve Kirby and Nancy Wang

# Disclaimer

The views in this presentation are those of the authors. Any errors or omissions are solely the responsibility of the authors.

# Course Outline

This is part 1 of a four-part course.

1. From Source Data to SDTM
2. From SDTM to “Analyzable” ADaM
3. From ADaM to Tables and Figures
4. Building Define.xml Data Documentation into the SDTM/ADaM Mapping Process

# Intended Audience

This course is primarily designed for scientists and programmers who have a basic understanding of SDTM structure. It uses SAS programming as the data transformation software; the techniques shown can be applied using other applications.

# Objective

To share some simple, effective strategies (with examples) for converting data as collected into CDISC SDTM format

# Assumptions

- Source data available in SAS format (labels preferred)
- Annotated CRF available that documents the dataset and variable names
- Mapping from source to SDTM done using SAS code

# Road Map

- **What Goes Where?:** Document mapping associations in metadata that drives the mapping process
- **What's your function?:** Plan programming based on variable function
- **Let's Go!:** Implement mapping strategy in metadata driven process
- **Make it Just So:** Review, UAT and close out

# Document Mapping Associations: Overview

**Goal:** Combine source data and documentation and SDTM information into content that drives the mapping process



# What Goes Where: Collect Source Information

- Collect a “data” overview of the source information.
  - Get “source data” annotated CRF
  - Generate SAS contents of “source data” in useful format (don’t neglect non-CRF data such as PK concentrations, lab results)
  - Get SAP, protocol, end user feedback

[Source\\_aCRF\\_EG.pdf](#)

[contents\\_source\\_lb.doc](#)

# What Goes Where: Specify SDTM Mapping with Annotated CRF

- Annotated CRF
  - Pictorial representation easily digestible
  - Gaps with non-CRF data
  - Gaps with data not exact match

[SDTM\\_aCRF\\_EG.pdf](#)

# What Goes Where: SDTM Specifications

- SDTM specifications are a (nearly) complete reference for standard domain and variable characteristics:
  - Names
  - Labels
  - Character/Numeric
  - Variable function
  - [SDTM\\_312\\_Metadata.xls](#)

# What Goes Where: Specify Mapping in SDTM Specs

- Augmented SDTM specs
  - Fewer gaps, less accessible than aCRF
  - This general approach can be used for any data structure where specifications are available
  - Can be expanded to fully support define.xml generation
  - **Use them** to drive mapping and to create “perfect” blank data

[sdtm\\_312\\_structure\\_lb.xls](#)

# What Goes Where: Code and Code Citations

[SPECS\\_CODE.sas](#)

Bradford J. Danner, Exploitation of Metadata for Restructuring Datasets and Code Generation, paper cc21, PharmaSUG 2008

David Grey, Zhuo Chen, Automated Verification of Dataset Metadata to Data, paper cc14, PharmaSUG 2009

Misha Rittmann, Automating the Link Between Metadata and Analysis Datasets, paper AD16, PharmaSUG 2010

# What Goes Where: Relevant for Submission?

- Strong bias in favor of submitting all information, but take a look if data do not seem to add value:
  - “duplicate” information
  - monitoring questions, pages (“Did subject show up, did subject have test”)
- Document, document!
- Avoid at collection where possible



Questions?

# What's Your Function: Overview

- All clinical study data can be usefully put into functional categories
  - Inform mapping decisions
  - Identify commonalities across domains
  - Inform code generation
  - Assists global standardization
- SDTM Guidance categorizes variables



# What's Your Function: Reuse and Recycle Code

- Look at function and find ways to use the same (or similar) code in many programs.
  - Enhances consistency
  - Global updates are easier
  - More efficient
  - Functional guidance from SDTM:

[sdtm\\_312\\_structure\\_lb\\_varcat.xls](#)

# What's Your Function: Reuse and Recycle Rules of Thumb

- If mapping stable across domains, consider global macro, metadata
  - Timing variables and Identifiers
- If mapping not very stable across domains, consider local macro, local metadata, local code
  - Topic, result, record, and synonym qualifiers

# What's Your Function: Across Domain, ID and Date/Time

- Identifiers
  - Company/submission standard  
[USUB.sas](#)
- Timing
  - ISO 8601
  - Careful with missing information  
[DATETIME.sas](#)

# What's Your Function: Across Domain, Scheduled Visit

- Visit (day level) information
  - Protocol references and Consistency across studies both important
  - Consider standardizing (within submission set, therapeutic area)
  - Trigger visits on information available in all data sets (scheduled timing, collection dates)
  - Pay attention to unscheduled visits
  - [VISIT.sas](#)

# What's Your Function: Across Domain, Scheduled Hour

- Time point (often hour) level information
  - Time from what? (keep it versatile, can vary by domain, needs to be consistent with other data)
  - Input data choice matters
  - Not always needed
  - [TPT.sas](#)

# What's Your Function: Within Domain Metadata

- Domain specific variables, especially topics and result qualifiers, now subject to defined terms
  - Targeted metadata can streamline processes and be reused across studies
  - Global mapping conventions can work well too
  - [SDTM\\_LB\\_TEST.xls](#)

# What's Your Function: SDTM “Leftovers” (CO and SUPP--)

- General comments
  - Consider how closely related to specific domain
  - If very close, consider SUPP--
- Supplemental information
  - Make sure it does not fit in main domain
  - Balance potential duplication and loss of useful information
  - Consider traceability



Questions?



# Let's Go: Structural Organization

- Program names
  - Naming by role/output
  - Modular approach enhances flexibility
- Folder structure (code and data)
  - Separate code by type/function
  - Permanent data location(s)
  - Interactions with end-users

# Let's Go: Process Roadmap

- Pull together relevant source data, augment
- Create full set of blank SDTM data using metadata and append to source data
- Call macros to create across domain variables
- Write code (and/or apply metadata) for within domain variables
- Keep variables as needed
- Label and output permanent data

# Let's Go: Relevant Source Data (and Prep)

Simple LB input data and data prep overview:

[LB\\_Program\\_Source\\_Input.doc](#)

[MERGE\\_ALL.sas](#)

# Let's Go:

## Within Domain Variables

- LB Specific Code Example (testcode metadata use):
  - Raw file format

[SDTM\\_LB\\_TEST.xls](#)

- Used (as SAS data) in program

[LB\\_Program\\_TESTCD.doc](#)

**Let's Go:**

## **Order Data, Keep What You Need and Output**

Order data based on metadata

Keep required, expected and (as needed) permissive

[LB\\_Program\\_Output.doc](#)

# Let's Go:

## Keeping Track of the “Leftovers”, SUPP–

- Safe and simple to build from SDTM pre-output “work” data
- Retain source information that adds value
- Resist derivations and duplications
- Label based on source documents
  
- [LB\\_Program\\_SUPP.doc](#)

## Let's Go:

# Keeping Track of the “Leftovers”, CO

- If comments connected to input data used for main domain, then same general process as SUPP—; but as comments go across data sets:
  - Output individual domain comments content
  - Combine in CO in separate step
- Worth considering if comments better placed in SUPP— as more value to reviewer

# Make It Just So: Review Categories

- Structural review – do data conform to SDTM standards?
- Substantive Review – do data give a full and accurate picture of the source
- Don't forget to document!



# Make It Just So: Structural Accuracy

- Variable characteristics built into metadata; SDTM rules built into code.
- Check details programmatically (OPEN CDISC, Web SDTM, SAS DDI)
- [websdm-to-sdtm.xls](#)

# Make It Just So: Substantive Accuracy

- Compare data as mapped to source
- Review considerations vary with type of mapping
  - 1 to 1, dates/times
  - Assigned, derived
- Simple, useful no-duplicate code can highlight all unique combinations
  - [lb\\_testcode.lst](#)

# Make It Just So: End User Feedback

- Does it support the SAP/Listing shells?
- How does it work for the people doing the listings?
- Involve end users in the process and ask for feedback.

# Make It Just So: Close Out

- Internal review documentation
  - Peer review
  - Program reports
- External (and or end user) review documentation
- Final data (and any data documentation) in a standard area
- Consider locking down final content and code

# Thanks!

Thanks to all the programmers, reviewers and end users who have helped and taken an active interest in making the CDISC standards work.



Questions?