

An aerial photograph of a city, likely Chapel Hill, North Carolina, showing a river and a large lake. The text is overlaid on this image.

Metadata 101: A Beginner's Guide to Table-Driven Applications Programming

Frank DiIorio

**CodeCrafters, Inc.
Chapel Hill NC**

This Presentation ...

- Describes construction of metadata-based systems, focusing on the **metadata** itself and the **tools** used to access it
- Summarizes benefits of, rules for usage
- Presents examples showing:
 - How a single metadata table can be used throughout the life cycle of a project
 - Benefits of centralized change
- Based on the above, concludes with “Reasons to Use Metadata” and “Lessons Learned”

Just What *Is* Metadata?

- Classically, “data about data”
- More broadly, data that can be used to describe a collection of related entities (data, directory structure, report layouts and other specifications, etc.)
- You already have SAS metadata: SAS Dictionary Tables and Views
- “Home grown” metadata complements the Dictionary Tables
- What we **don't** discuss here is the SAS Metadata server

Understanding the Need for Metadata (1 of 2)

Consider work flow for a project (pharmaceutical, but the general points hold up):

- Specification of deliverables
- Programming / validation
 - Datasets
 - Reports (Tables, Listings, Figures – aka "TLFs")
- Deliverables to client
 - Datasets
 - Reports
 - Supporting documentation

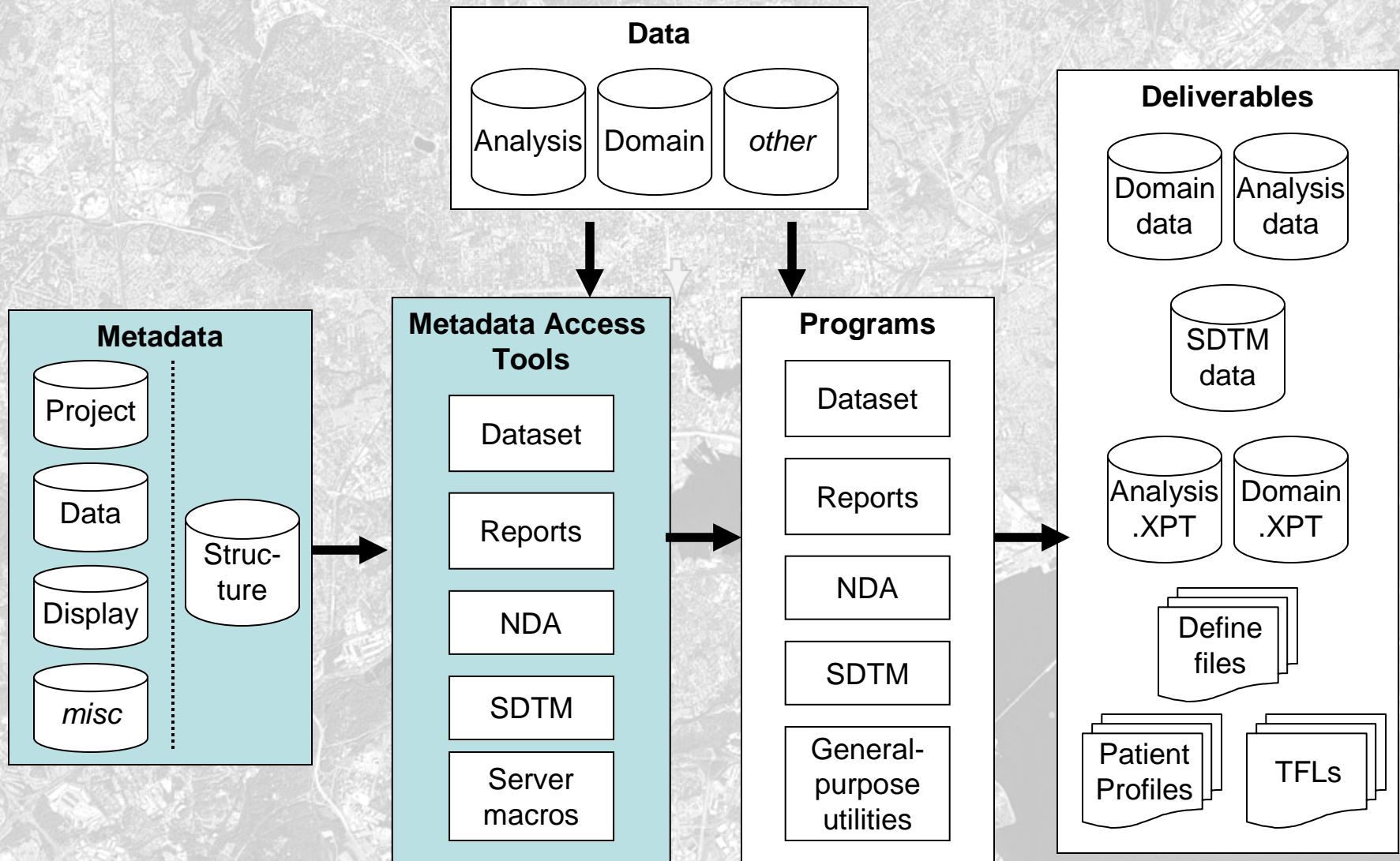
How are these typically created?

- Spec's in Word or Excel doc
- Transcription / cut & paste of variable names, titles and footnotes, etc. into programs

Understanding the Need for Metadata (2 of 2)

- This can be problematic
 - Spec-program can get out of sync
 - Tedious, error-prone
 - Time consuming, especially when deadlines are approaching
- Partial solution ... move specs, program fragments into programmatically-accessible data stores (i.e., metadata)
- **Start developing metadata-driven applications by identifying hard-coded and/or repeated items and moving them out of programs and into tables/datasets.**
- Make the metadata easily accessible to those who need it (programmers, end users). Develop metadata access tools.
- Big Picture follows ...

Use Metadata and Metadata Access Tools



Big Picture: Initial Comments

- **Contents evolve.** Initially, just a few tables. Number and content change in order to meet user needs.
- **Tools are critical.** Macros provide easy access to sometimes complex data structures.
- **Interface** for data entry is important. ([view](#))
- **Tables replace some parts of work flow** and introduce new possibilities for automation. A single table can be used throughout the life cycle of a project
- **This is not intuitive.** Most of what we learned has been trial and error. A forgiving work environment really helps!

Examples will demonstrate these points. [next](#)

Interface (1 of 2)



CDISC Metadata
Editor Version 1.0

Domains

Domain	<input type="text" value="AE"/>	Class	<input type="text" value="Events"/>	Description	<input type="text" value="Adverse Events"/>
Structure	<input type="text" value="One record per event per subject"/>			Purpose	<input type="text" value="Tabulation"/>
Notes	<input type="text"/>			Footnotes	<input type="text" value="* indicates variable may be subject to sponsor-controlled terminology~ ** indicates variable may be subject to external-controlled terminology."/>
Input Notes	<input type="text"/>			Output Notes	<input type="text"/>
Keys	<input type="text" value="STUDYID, USUBJID, AETERM, AESTDTC"/>			Submit	<input type="button" value="Submit"/>

Note: the domain AE is a CDISC requirement. The fields for structure, class, purpose and description are locked and cannot be edited.

- Edit variables for this domain
- Main Menu
- Exit this Application

[back](#)

Interface (2 of 2)

Variables

CDISC Metadata Editor Version 1.0

Variables for Standard CDISC Domain: AE

Non Editable Fields*

Name	AEACN
Domain	AE
Label	Action Taken with Study Treatment
Core	Exp
Role	Record Qualifier
Sequence	18
Type	Char

Length Origin

Source

CRF Page Name

CRF Page Number

CT

Submit?

Programming Definition

Notes

CDISC Notes Describes changes to the study treatment as a result of the event. Examples include ICH E2B values: DRUG WITHDRAWN, DOSE REDUCED, DOSE INCREASED, DOSE NOT CHANGED, UNKNOWN or NOT APPLICABLE.

FDA Definition

*Attributes defined by CDISC standard for domain AE - cannot be modified.

Record: of 39

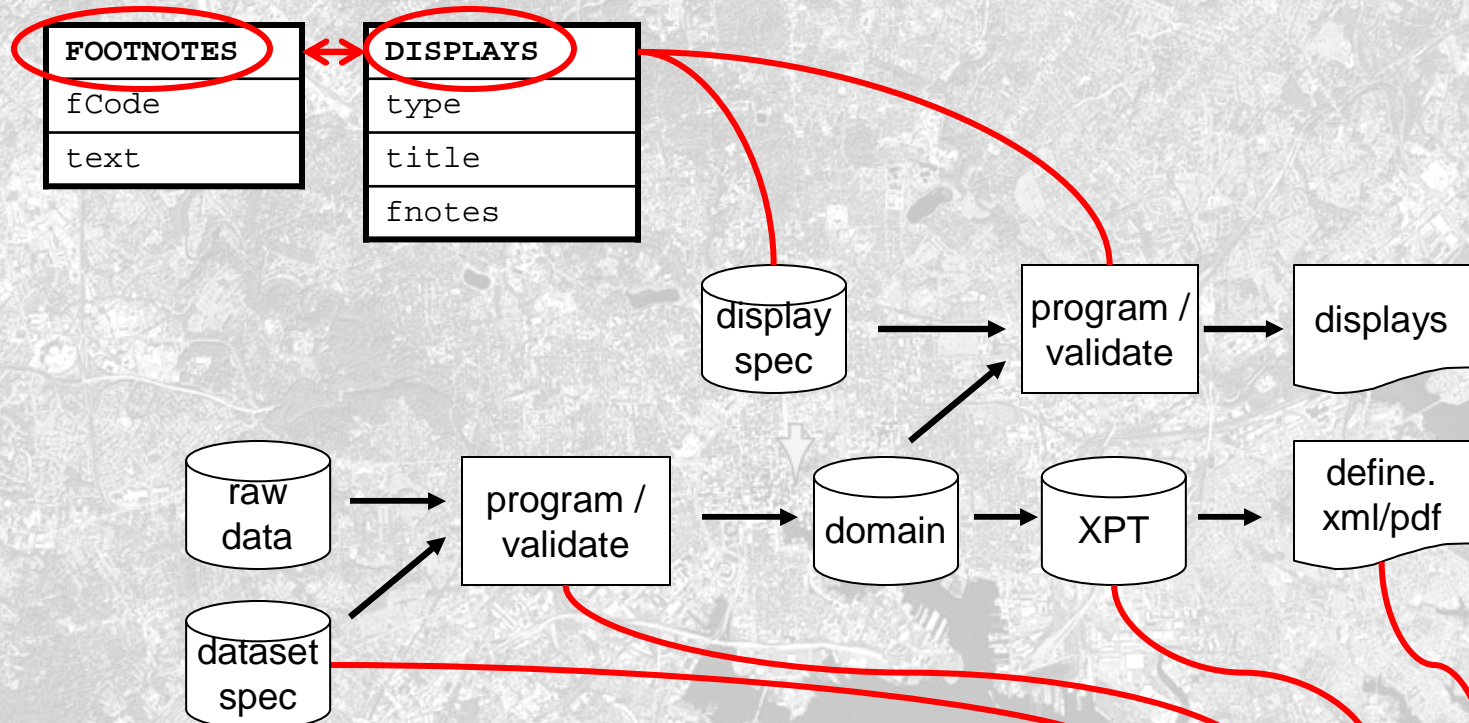
[intro](#)

[back](#)

Examples

- The power of metadata-driven systems is best seen through examples
- There are many tables used throughout the life of even simple projects.
- Here, though, we'll focus on two examples.
 - Display-related: as an example of a specialized table that replaces a significant amount of previous hard-coding, resulting in improved quality and speed of production of output
 - Variable-level: as an example of a table that's used in many places
- Next slide shows contents of the metadata and how the tables are used throughout the life of the project

Use of Variable and Display-Related Tables



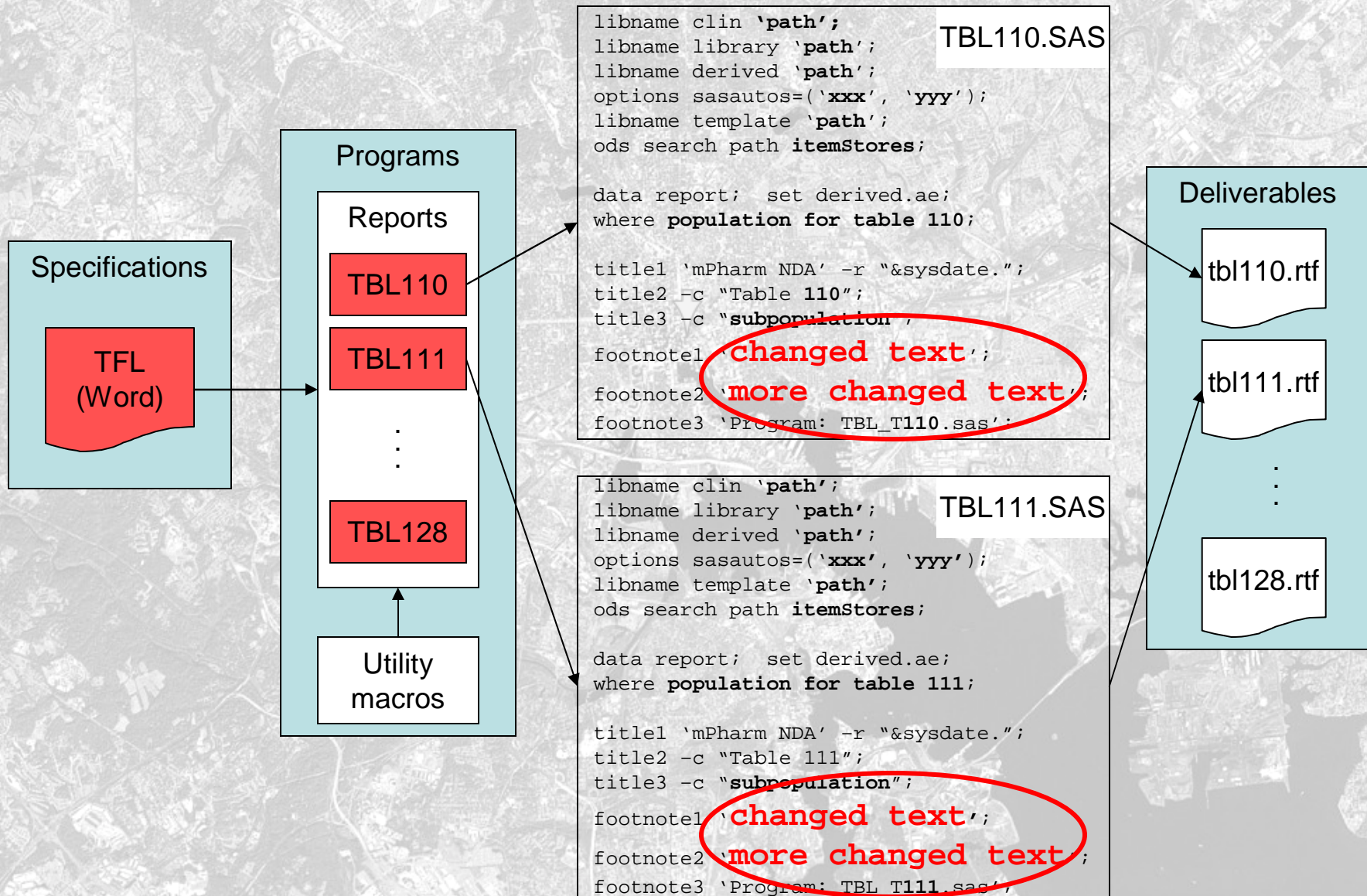
FOOTNOTES	DISPLAYS
fCode	type
text	title
	fnotes

VARIABLES	dataset	Dataset name	✓	✓	✓	✓
	variable	Variable name	✓	✓	✓	✓
	type	Data type (c, n)	✓	✓		✓
	length	Var. length	✓	✓		
	label	Variable label	✓	✓		✓
	definition	Derivation or source	✓			✓
	sequence	Order in output dataset	✓	✓	✓	
	include	Include in output dataset?	✓	✓	✓	
	docLoc	If not derived, source doc page name, number				✓

Example 1: Displays

- Without metadata
 - Spec doc identified input dataset, display type, identifier, titles, footnotes
 - Manually enter these into the program
- Problems
 - Tedious, error-prone
 - Repetitive
 - Changes become difficult to implement: change in spec affects “n” tables. How can you be sure you change the correct “n”, rather than “not n”?
- Example
 - Change in footnote affects 10 tables
 - Note that text is changed 11 times: once in spec document, 10 times (hopefully!) in the programs

Example 1: Displays, Pre-Metadata Footnote Text Change



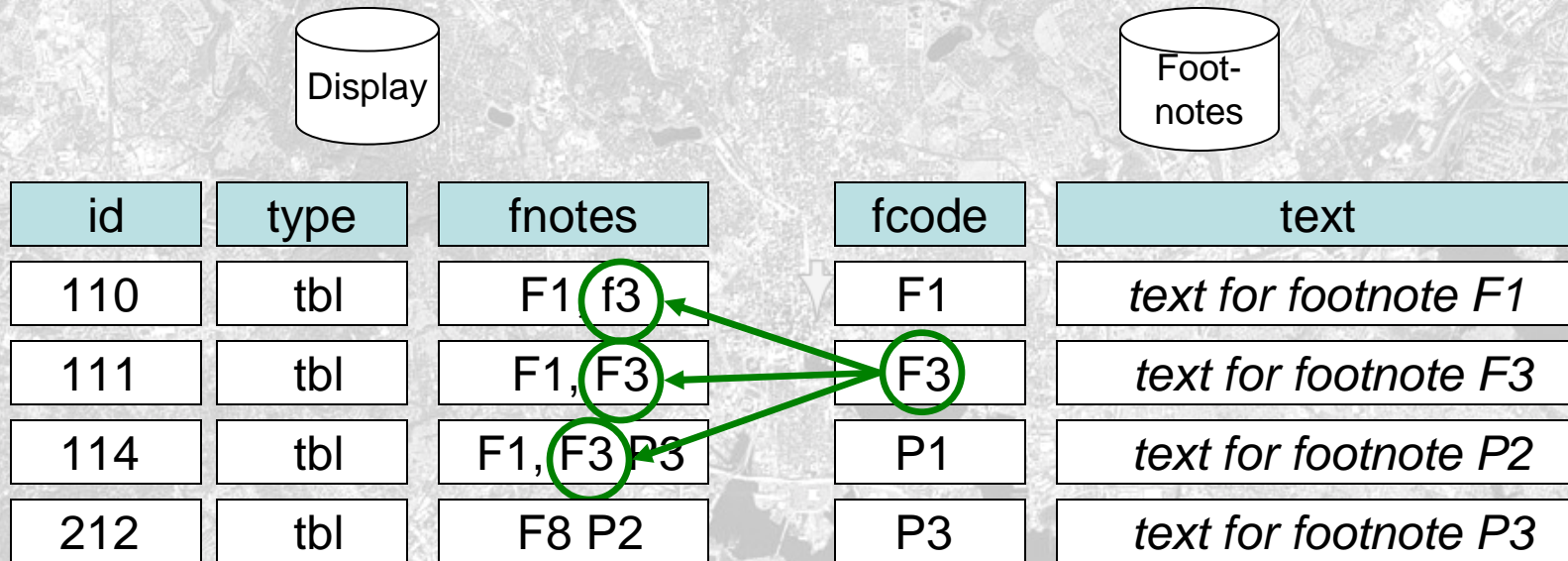
Example 1: Displays, Metadata-Oriented Solution

- Recall the problem: how to reliably propagate footnote text change to programs using the footnote
- Develop metadata:
 - Ask "what is repeated in multiple locations and can be moved into metadata?" This identifies candidates for inclusion in metadata. Footnote text can be moved into a table.
 - Insert footnote codes in `DISPLAY` table
 - Enter footnote code, text in `FOOTNOTE` table
- Develop tools to simplify metadata access:
 - Pass display type and identifier to `%TFL`
- Use the tool in the display program
 - Call `%TFL`
 - Use the macro variables that it creates
 - Educate the programmers about its use

This is illustrated in the next slide ...

Example 1: Displays - Metadata-Oriented Solution (cont.)

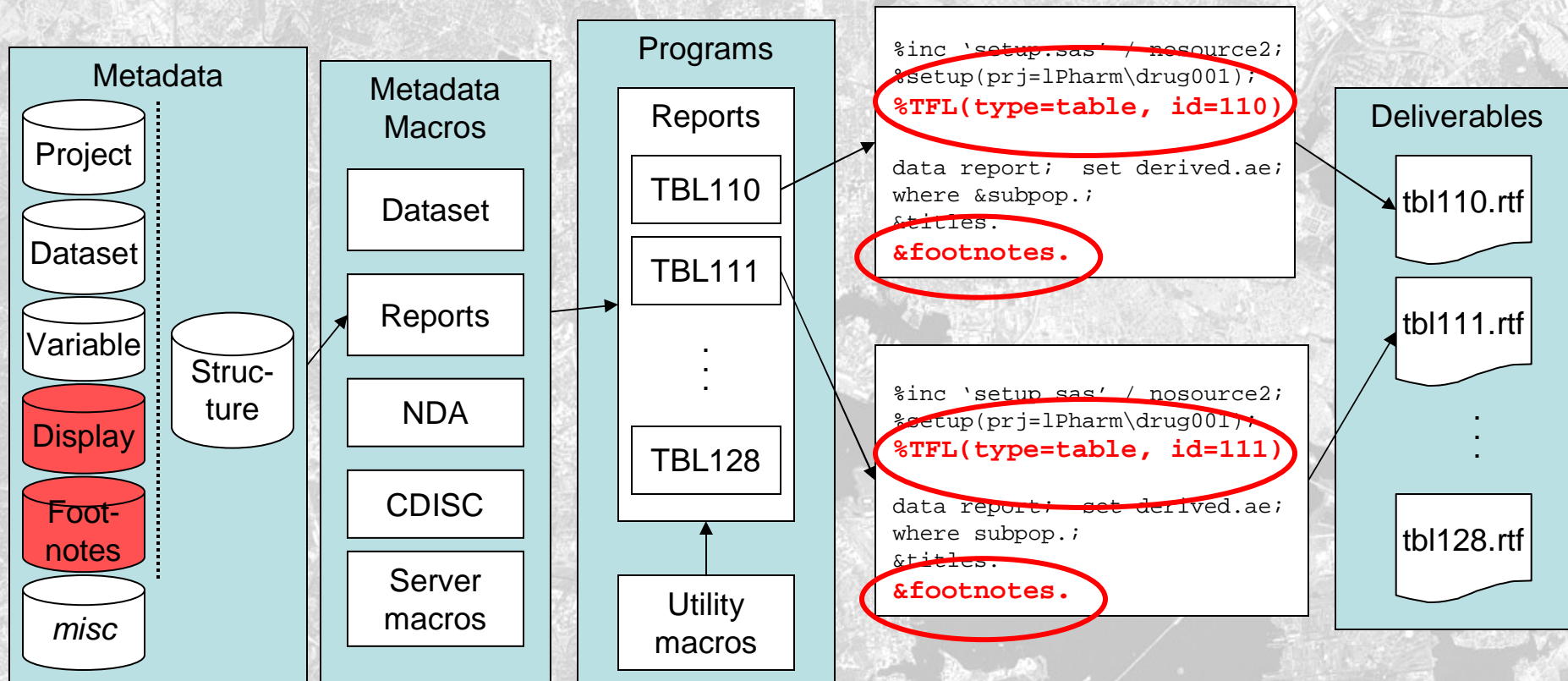
First, create DISPLAY and FOOTNOTES metadata tables.



Consider the impact of a change in the FOOTNOTES table for FCODE = F3. We need a way to get the new text to the programs that are writing tables using footnote code F3.

We develop a tool – %TFL (next slide) – to complement the metadata, making it easy to use.

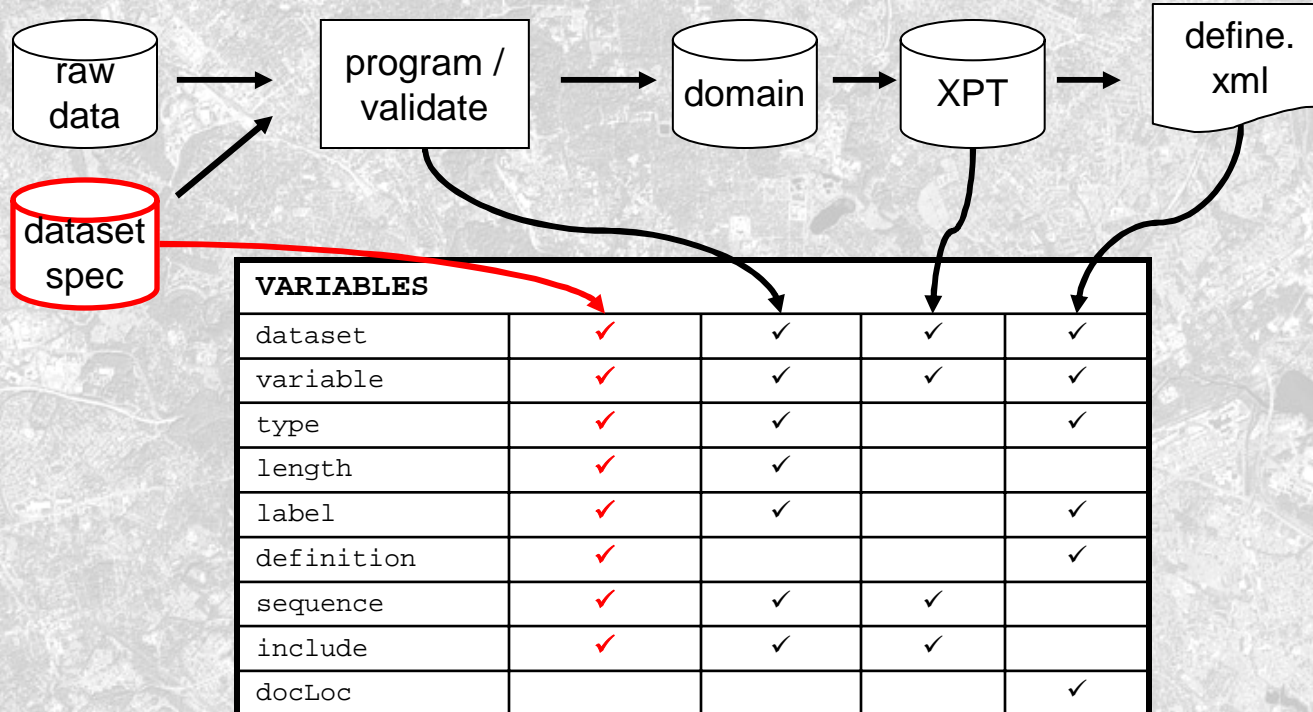
Example 1: Displays - Metadata-Oriented Solution (cont.)



Example 2: Use Throughout the Project

- Review simplified project life cycle for a set of deliverables to FDA:
 - dataset specifications
 - program and validate
 - convert to transport format
 - documentation (PDF and/or XML)
- We'll look at each phase, focusing on:
 - metadata v. non-metadata programming
 - use of different fields in VARIABLES metadata at different phases

Example 2: Dataset Spec (1 of 2)



No metadata

Print Word document (select datasets of interest; sort by name or position in the dataset)

Repeat selection, sort, print process each time a dataset changes

Utilize Metadata

`%printSpecs`

Reads VARIABLES table for datasets of interest; can select only recently-edited variables; display by variable name or sort order; select level of detail to display

`%printSpecs(use=ae cm, order=name, since=-2)`

Example 2: Dataset Spec (2 of 2)

Variable Specs for Domain AE

Gold Standard [N:\submissions\metadata\SDTM\11.mdy - Study ██████████.SDTM with Mod Date 13FEB07:14:21]

Key vars from DOMAINS table: STUDYID USUBJID AETERM AESTDTC

Variables are arranged by their position in the output Domain. Only Expected/Required and SUBMITDB=Yes Variables are Displayed

2

							CRF Page		
CDname [2]	Label [1]	Type [1]	Len	Core [1]	sub mit DB	Source	Name	#	Comments
STUDYID	Study Identifier	Char	12	Req	Yes	CRF	DEMOGRAPHICS AND REPRODUCTIVE STATUS	1	[ProgDef] ='E7974A00110Z'
DOMAIN	Domain Abbreviation	Char	2	Req	Yes	DER			[ProgDef] ='AE'
USUBJID	Unique Subject Identifier	Char	24	Req	Yes	DER			[ProgDef] Concatenate RAW.DEMO.PROTOCOL and three-digit RAW.DEMO.ID separated by '-'
AESEQ	Sequence Number	Num	4	Req	Yes	MC			[ProgDef] -RAW.AECDISC.AESEQ
AETERM	Reported Term for the Adverse Event	Char	70	Req	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AE
AEDCOD	Dictionary-Derived Term	Char	200	Req	Yes	DER			[ProgDef] -RAW.AECDISC.PT
AESODSYS	Body System or Organ Class	Char	200	Exp	Yes	DER			[ProgDef] -RAW.AECDISC.SOCT
AESEV	Severity/Intensity	Char	8	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -pu\$RAW.AECDISC.AE_SEV, AESEVF.)
AESER	Serious Event	Char	3	Exp	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -pu\$RAW.AECDISC.SERIOUS, NYF.)
AEACN	Action Taken with Study Treatment	Char	32	Exp	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -pu\$RAW.AECDISC.ACTION, ACTF.)
AEACNOTH	Other Action Taken	Char	100	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AEACNOTH
AEREL	Causality	Char	18	Exp	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -pu\$RAW.AECDISC.AE_REL, AERELF.)
AEOUT	Outcome of Adverse Event	Char	25	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -pu\$RAW.AECDISC.OUTCOME, AEOUTF.)
AESCONG	Congenital Anomaly or Birth Defect	Char	1	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AESCONG
AESDISAB	Partial or Significant Disability/Incapacity	Char	1	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AESDISAB
AESDTH	Results in Death	Char	1	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AESDTH
AESHOSP	Requires or Prolongs Hospitalization	Char	1	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AESHOSP
AESLIFE	Is Life Threatening	Char	1	Perm	Yes	CRF	ADVERSE EXPERIENCES	03	[ProgDef] -RAW.AECDISC.AESLIFE

[1] Red background indicates a Gold Standard that was changed at the study level. The value displayed is the Study value.

[2] Blue background indicates Key variable in DOMAIN variable KEYS

Table entries: [1] Required/Expected variables and/or [2] CDNAME found in project metadata

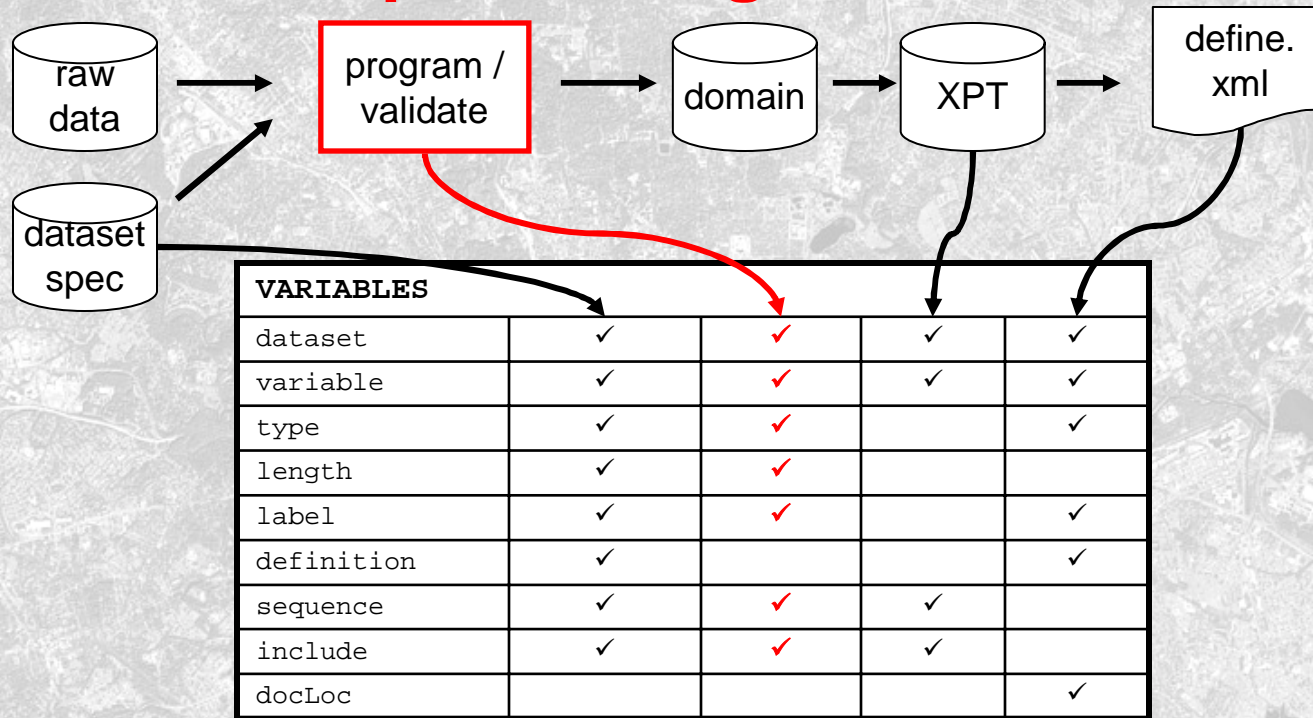
Shaded rows indicate Required or Expected variables (CORE='Req' or 'Exp')

Run date-time: Wednesday, February 14, 2007 2:19 PM

Program: ██████████.sas

back

Example 2: Program / Validate



No metadata

Manually code or cut and paste labels and other variable attributes into the dataset creation program.

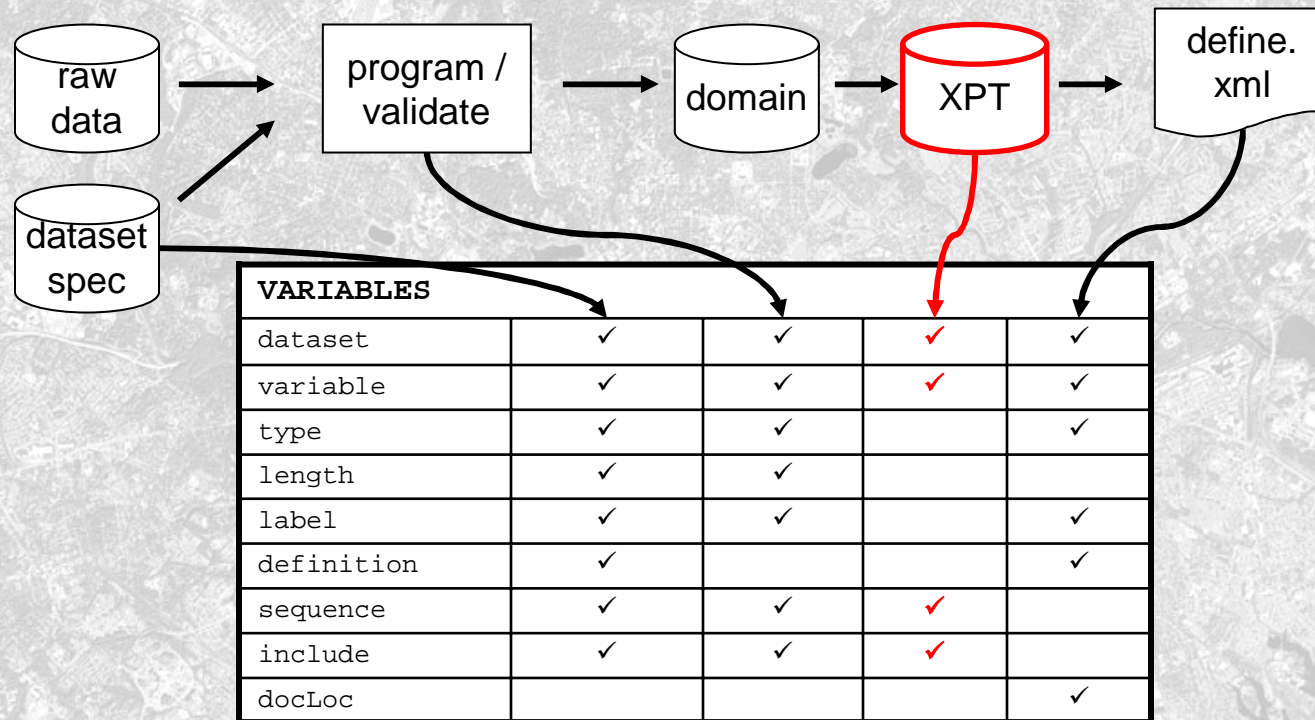
Make sure that changes to attributes and variable order in the spec are reflected in the program.

Utilize Metadata

%attrib macro reads metadata and creates KEEP and ATTRIB statements. Only variables marked for inclusion are included; they are in the correct order.

```
%attrib(data=ae)
data dom.ae;
set work.ae;
keep &keep.;
attrib &attrib.;
run;
```

Example 2: Export



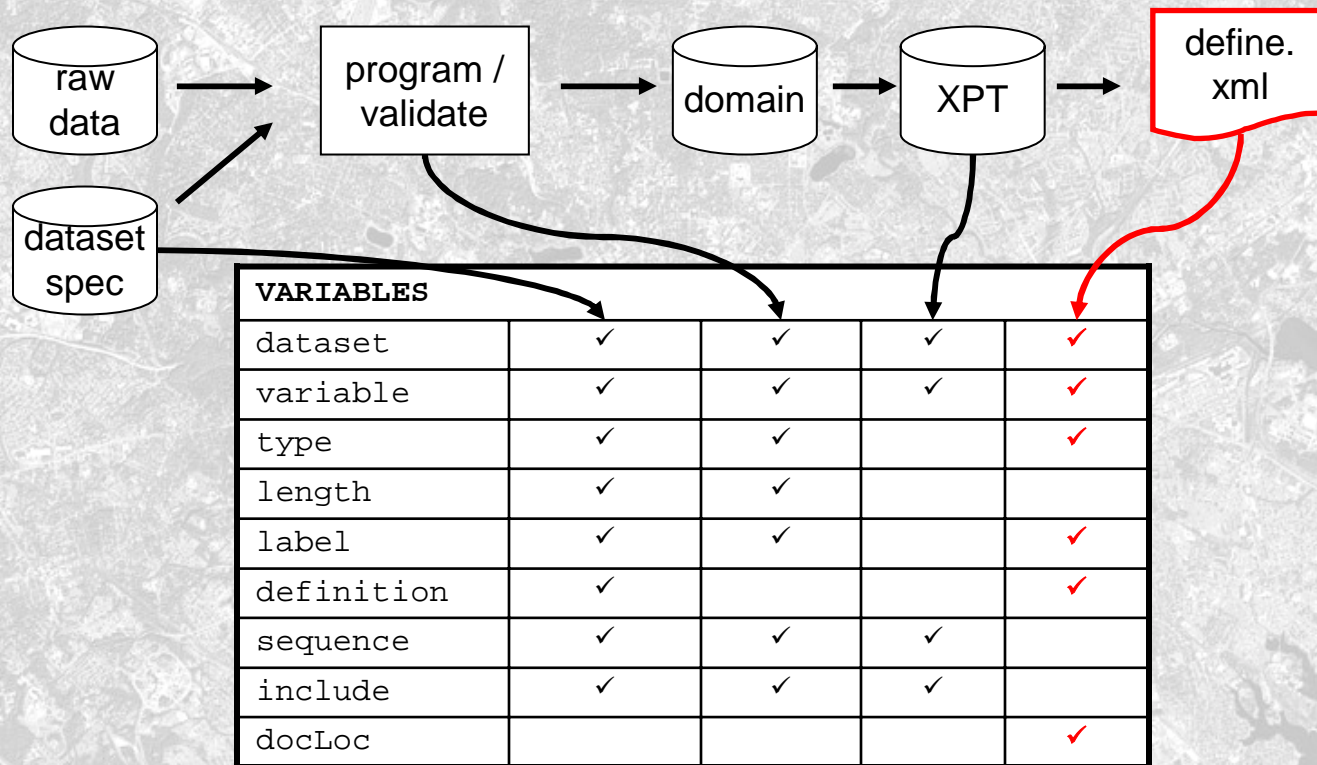
No metadata

Manually coding. Retrieve file size of input dataset and determine how many 100MB “chunks” will be required; ensure that dataset attributes (type, length, variable order) match those of input data.

Utilize Metadata

%export macro reads metadata, SAS Dictionary tables to perform final QA (attribute comparisons of input Domain and output XPT).

Example 2: Documentation - define.XML/PDF (1 of 2)



No metadata

Hard to imagine doing this in a non-automated way, but you could ...

Edit Word dataset spec document; manually check that variable attributes in the XPT datasets; convert to XML with style sheet (XSLT file) that presents documentation in preferred format.

Utilize Metadata

%define

Example 2: Documentation - define.XML/PDF (2 of 2)

Study: 1001 Protocol: BlockBuster Drug Description: BluesBGone

Document generated 2007-09-14T12:40:27

[Analysis Datasets](#) [Variables](#) [Value Metadata](#) Results: [Figures](#) [Listings](#) [Tables](#)

34 Domain Datasets

[Print](#) [Analysis Datasets](#)

Dataset	Description	Structure	Purpose	Keys	Location
AE	Adverse Events	One record per event per subject	Tabulation	STUDYID, USUBJID, AETERM, AESTDTC	AE.xpt
CM	Concomitant Meds	One record per medication intervention episode per subject	Tabulation	STUDYID, USUBJID, CMTRT, CMSTDTC	CM.xpt
CO	Comments	One record per comment per subject	Tabulation	STUDYID, USUBJID, COSEQ	CO.xpt
DM	Demographics	One record per subject	Tabulation	STUDYID, USUBJID	DM.xpt
DS	Disposition	One record per disposition status or protocol milestone per subject	Tabulation	STUDYID, USUBJID, DSSTDTC	DS.xpt
DV	Violations/Deviations	One record per violation/deviation per subject	Tabulation	STUDYID, USUBJID, DVSEQ	DV.xpt
EG	ECG	One record per ECG observation per time point per visit per subject	Tabulation	STUDYID, USUBJID, EGTESTCD, VISITNUM	EG.xpt
EX	Exposure	One record per constant dosing interval per subject	Tabulation	STUDYID, USUBJID, EXTRT, EXSTDTC	EX.xpt
IE	Inclusion/Exclusion Exceptions	One record per ind/excl criteria exception per subject	Tabulation	STUDYID, USUBJID, IETESTCD	IE.xpt
LB	Labs	One record per lab test per time point per visit per subject	Tabulation	STUDYID, USUBJID, LBTESTCD, VISITNUM, LBSTCD	LB.xpt
MH	Medical History	One record per medical history event per subject	Tabulation	STUDYID, USUBJID, MHTERM	MH.xpt
NE	Nasal Exam	One record per nasal exam or nasal abnormality per visit per subject	Tabulation	STUDYID, USUBJID, NETESTCD, VISITNUM	NE.xpt
PC	PK Concentration	One record per time point per subject	Tabulation	STUDYID, USUBJID, PCTESTCD, VISITNUM, PCTPT	PC.xpt

[back](#)

Summary: Reasons to Use Metadata

- Less end-user coding
- Improved reliability of outputs
- Happier programmers
- It's cool and cutting-edge – happier developers
- Elimination of duplicate coding
- Faster application development
- Metadata tables can be “repurposed”
- Improves corporate responsiveness to clients (can provide different formats of deliverables more easily than hard-coded solutions)
- Integrity of specs is improved (database v. .doc, .xls files)
- Flexible architecture is amenable to new metadata, new tools, further process improvements

Summary: Lessons Learned

- Recognize the importance of the user interface
- Realization that change is good
- Develop programmer tools
- Develop end-user tools
- Chose a robust database for metadata storage (multi-user; transparent, issue-free access to SAS)
- Utilize Dictionary Tables (view as complementary metadata)
- Train all users (part of new employee orientation)
- Conduct focus groups (solicit ideas for modified/new tools)
- Understand that tool change means you've become a software developer (SGF 2008 paper)

Contact

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

Frank@CodeCraftersInc.com

www.CodeCraftersInc.com