

Dimensions of Standards' Versioning: From Variables to Program Code

Astrid Wolter

- **Version Control Synonyms:**
 - Revision Control, Source Control, Versioning...
- **Definition:**
 - Wikipedia: “Management of changes...”
- **Paradigms:**
 - Lock-modify-unlock (e.g. Visual SourceSafe)
 - Copy-modify-merge (e.g. SVN, CVS, Git, ClearCase...)

Classic VCS are file-based!

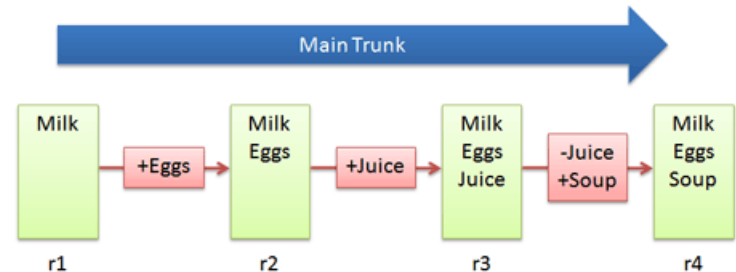
■ Concepts:

- Revision
- Release, minor/major version
- Head, trunk/main, branch
- Diff
- Conflict

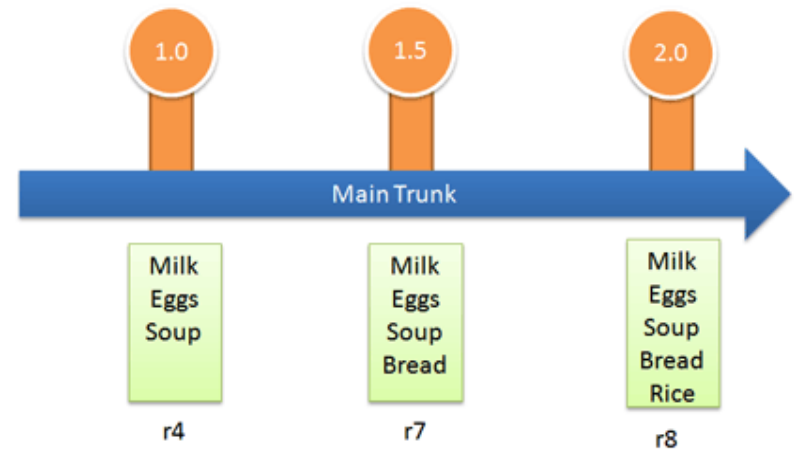
■ Actions:

- Add
- Check out (for edit)
- Check in
- Restore, revert version
- Version tree
- Tag, label
- Merge
- Locking, lock break...

Basic Diffs



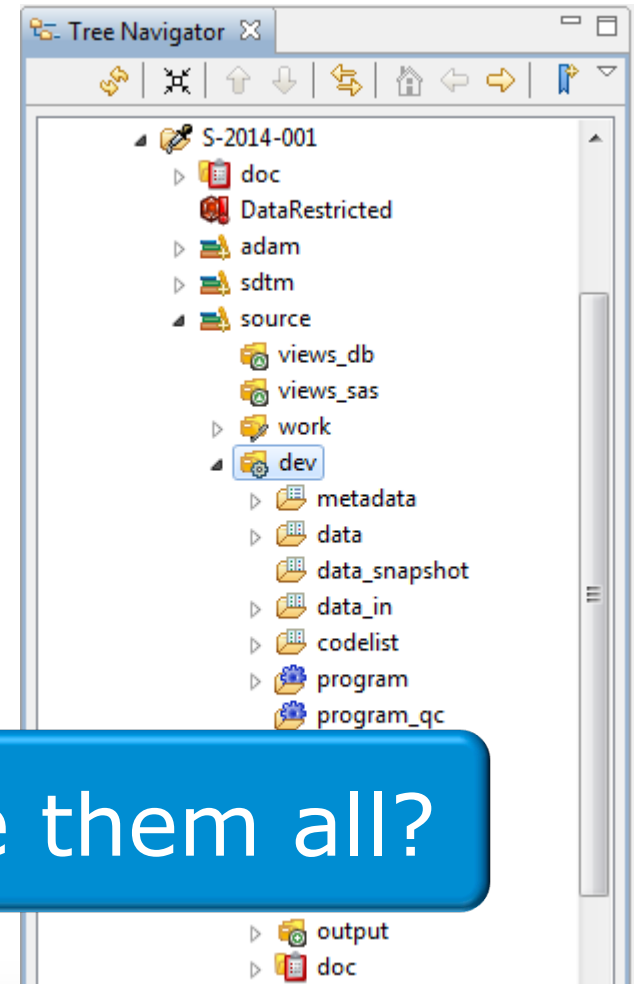
Tagging



- **Best practices in software engineering:**
 - Descriptive commit message
 - Make each commit a logical unit
 - Incorporate changes frequently
 - Coordinate with your co-workers
 - Never refill/re-justify paragraphs
 - Don't use long lines
 - Don't commit generated files
 - Understand your version control tool...

Is pharma world different?

- **entimICE repository content (sample):**
 - Data
 - Metadata
 - Codelists
 - Checks
 - Programs
 - Mappings
 - Outputs /TLFs
 - Logs..



Can classic VCS handle them all?

- **Data types:**
 - SAS datasets
 - Data warehouse tables
 - CSV data
 - XML data...
- **Challenges:**
 - Mixed storage types (files and database)
 - Small, big and very big data

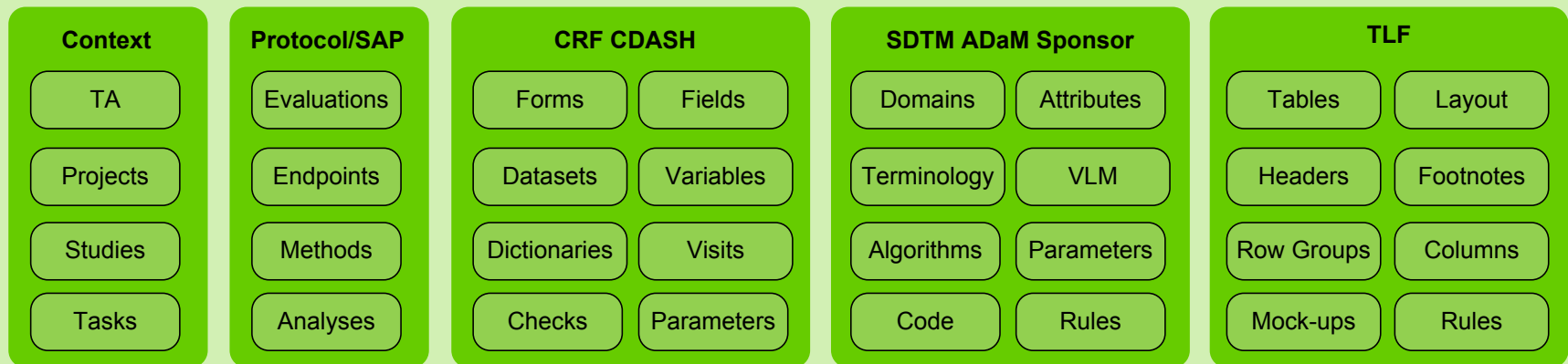
Classic version control tools are
not prepared for pharma data

■ End-to-End Standards:

- Tabular models (e.g. CDASH, RAVE, SDTM, ADaM...)
- Class- and concept-based models (e.g. BRIDG)
- Non-tabular standards (e.g. TLF layouts, titles, footers)
- Different metadata attributes per data model
- Global, project, study levels

entimICE End-to-End Standards World

entimICE Repository



■ Challenges:

- Each level in hierarchical data models might have own versions in addition to model versions

The screenshot displays a software interface with two main panels. The left panel, titled 'Tree Navigator', shows a hierarchical tree structure of SDTM models. The tree starts with 'sdm' at the root, followed by '3_1', '3_1_1', '3_1_2', '3_1_2a', '3_1_3', and '3_2'. Under '3_2', there are 'prod' and 'dev' folders. The 'dev' folder contains a 'cde' folder, which in turn contains 'ae', 'ce', 'cm', and 'co' folders. The 'co' folder is expanded, showing a list of tables: CODTC, COEVAL, COREF, COSEQ, COVAL, and DOMAIN. The right panel, titled 'co', displays metadata for the selected 'co' folder. It includes fields for Source, Locked by, Created, Action, Status, and Comment. The 'Created' field shows '02.09.15 17:22:31' and the 'Action' field shows 'Metadata updated'. The 'Status' field shows 'development'. Below these fields is a 'History' table with three columns: Creation Date, Action, and Version.

Creation Date	Action	Version
02.09.15 17:22:31	Metadata updated	1.4
26.05.15 18:54:13	Metadata updated	1.3
24.05.15 21:06:53	Metadata updated	1.2
15.01.15 11:58:01	Metadata updated	1.1
15.01.15 11:54:10	Batch extract of cde from tables	1.0

(Optional output with gray background)

TITLE_DRUG TITLE_INDICATION TITLE_TABLE_NO TITLE_AUTHOR1 TITLE_AUTHOR2 ddmmwwxx
PAGE page_no

Patient Validity Status and Reasons for Exclusions from Analysis [TITLE_CONTENTS1, 2, ..., 6]

Population: Randomized Patients [TITLE_POPULATION]

Pool: e.g. all controlled studies [TITLE_POOL]

	Treatment Group 1 (N=)	...	Treatment Group k (N=)	Total (N=)
Patients Randomized				n
Patients Valid for Safety				n(%)
Patients Valid for ITT Analysis				n(%)
Patients Valid for Per-Protocol				

5_VALIDITY_STATUS

parameters

TITLE_AUTHOR1
TITLE_AUTHOR2
TITLE_CONTENTS1
TITLE_DRUG
TITLE_INDICATION
TITLE_POPULATION
TITLE_TABLE_NO

titles

footers

FOOTNOTE1
FOOTNOTE2

sort_groups

log

result

program

views

FOOTNOTE2

Reporting Item

Details

VALUE

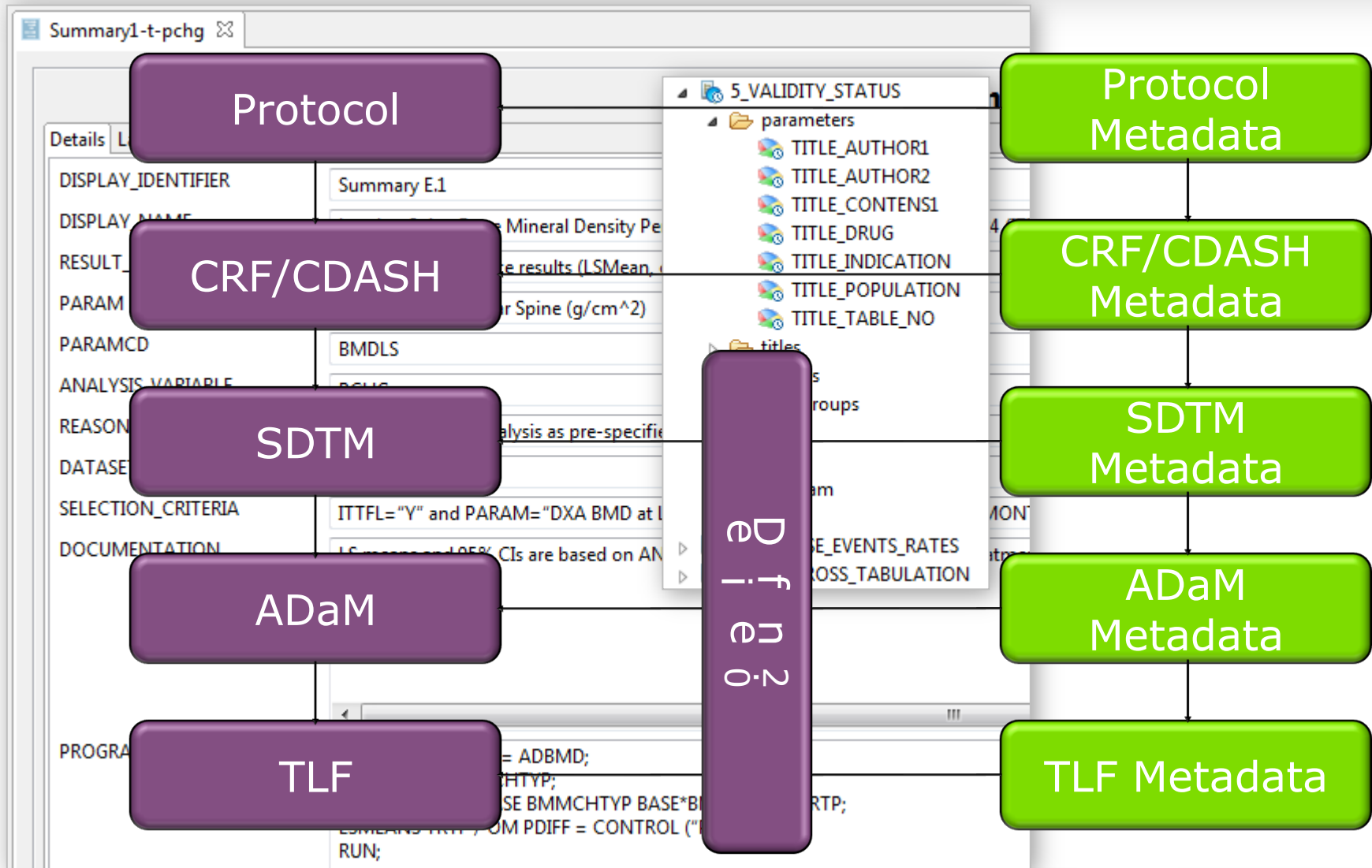
NOTE: THE REASONS ARE SORTED IN ALPHAB. ORDER.

DATA_DRIVEN

false

DATA_VARIABLE

CONDITION



- **Programming Artifacts:**

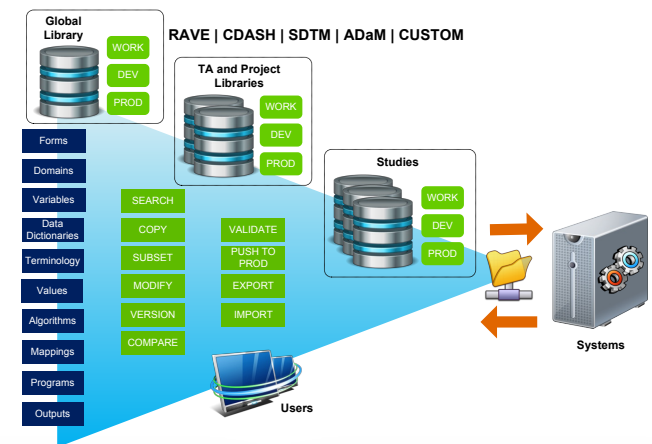
- Program code versions
- Run parameters
- Macros and versions
- Input data versions
- Outputs (e.g. ADaM data, TLF)
- Logs...

- **Challenges:**

- Traceability and reproducibility requirement goes beyond source code control!

■ Differentiate Work, Val, Prod areas

- Supports parallel work without locks
- No direct changes in prod – lifecycle goes always via val areas
- Different access levels for work/val/prod
- Commits – manual in val/work, auto in prod
- Area locks when lifecycle is finished



■ Combine physical and logical versions!

Logical versioning per data model with lifecycle

Physical versioning per object

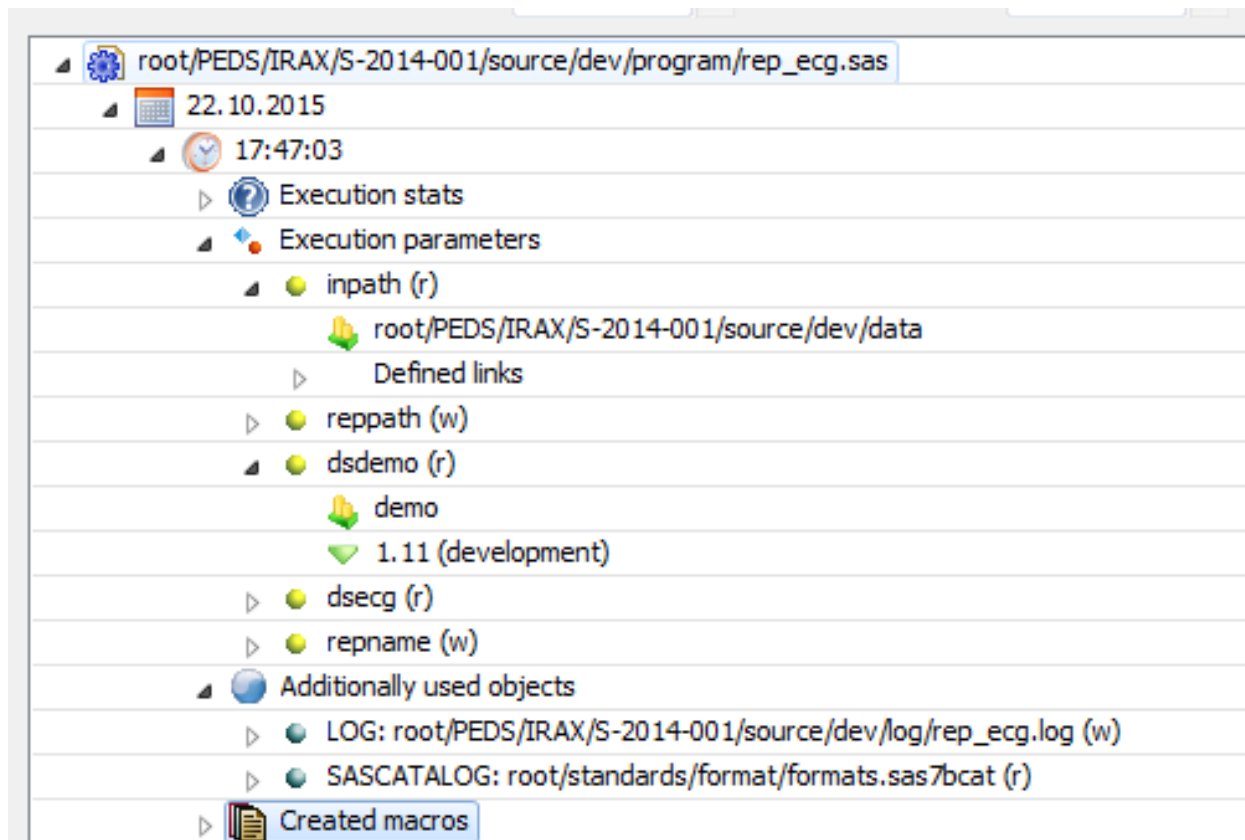
The screenshot displays the Entimo software interface. On the left, the 'Tree Navigator' shows a hierarchical structure with 'sdm' expanded, showing versions 3.1, 3.1.1, and 3.1.2. A blue arrow points from the 'Logical versioning per data model with lifecycle' text to this tree. On the right, the 'Select object version' table shows a list of versions with columns for Creation Date, Action, User, Version, and Status. A blue arrow points from the 'Physical versioning per object' text to the 'Version' column. Below the table, the 'Details' section shows the 'Model' set to 'SDTM' and 'Version' set to '3.1.2', with a blue arrow pointing from the 'Domains and variables contain model and version information; easy traceability across the repository through indexed search' text to these fields.

Creation Date	Action	User	Version	Status
07.09.15 14:54:07	Copied to prod after status change to Implemented	entjod01 (John Doe)	1.3	production (110)
07.09.15 12:02:23	Signed action: Workflow/Activate production accepted	entjod01 (John Doe)	1.2	production (110)
07.09.15 12:01:59	Signed action: Workflow/Retire accepted	entind01 (John Doe)	1.1	inactive (130)

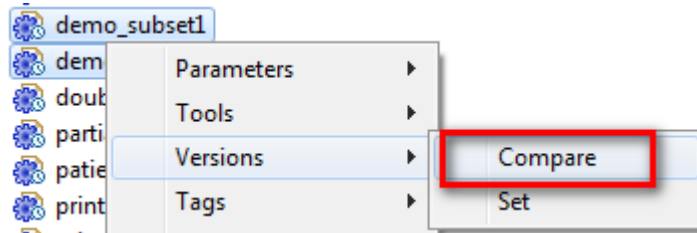
Details

Domain: XZ
Class: Findings
Class Order:
Model: SDTM
Version: 3.1.2
Repeating: Yes
Purpose: Tabulation
Structure: One record per finding
Is Reference?: No
Description: Surgical and diagnostic intervention

- **Combine versions and process tracing!**
 - Program execution metadata as evidence



■ Show differences for line-based content ...

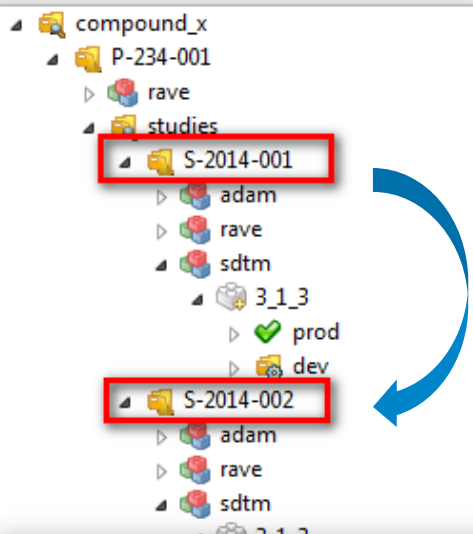


```
1 /*****
2 *****/
3 ***
4 *** Mapping/Pooling Program demo_subset1
5 ***
6 *** Generated on 09DEC2013:16:30:12
7 *** Using version 1.2
8 ***
9 *** Description:
10 *** Filter patients with id<20
11 ***
12 *** Consists of steps:
13 *** 1: step1 (Quick-Join Task)
14 ***
15 *** Uses external datasets:
16 *** S.&S_DEMO
17 *** S.&S_DEMO_OUT
18 ***
19 *****/
20 *****/
21
22 /*COMMENT1*/
23
24 %macro poolexec;
```

```
1 /*****
2 *****/
3 ***
4 *** Mapping/Pooling Program demo_subset2
5 ***
6 *** Generated on 09DEC2013:16:31:45
7 *** Using version 1.1
8 ***
9 *** Description:
10 *** age > 30
11 ***
12 *** Consists of steps:
13 *** 1: step1 (Quick-Join Task)
14 ***
15 *** Uses external datasets:
16 *** S.&S_DEMO
17 *** S.&S_DEMO_OUT
18 ***
19 *****/
20 *****/
21
22 %macro poolexec;
23
24 OPTIONS nofmterr;
```

■ ... and for binary content

Use case: Compare all metadata in study S-2014-001 and in S-2014-001, similar compare global and study levels



Terminology

dict CVTEST CVTESTCD		
General Aspects		
No Severe Problems		
Compare Keys and Attributes Source vs. Compare		
Problem_Category	Result	Problem
CDISC_Synonyms (Source) eq CDISC_Synonyms (Co...	ERROR	Source key(s)[code = C119245] values [Source: Cardiac Index XXX Compare: Cardiac Index]
Length (Source) eq Length (Compare) VIOLATED	ERROR	Source key(s)[code = C119245] values [Source: 3 Compare: 8]

- **Search capabilities shall include version-related information!**
- **Indexed Content:**
 - All textual Files (text and log files, also SAS programs...)
 - PDF files, Excel sheets, XML files...
 - SAS datasets, ClinRep datasets, CSV files as tables...
 - entimICE object metadata (e.g. variables, study metadata)
 - Version information and links
 - Process metadata (e.g. mappings and exec traces)
- **Search examples:**
 - fda AND cdisc
 - node #version in (1.7, 1.10)
 - node #version_history in (1.7, 1.10)
 - node #ext = sas7bdat AND oncology
 - node #mod (today -5 weeks, today -4 weeks)

- Classic version control tools and concepts are not sufficient for pharma processes!
- **Enhanced version control:**
 - Copy – modify – merge - lock
 - Management of logical and physical versions in parallel required
 - Version control for files and non-file content
- **Enhanced programming tools:**
 - Tagging for all content types
 - Diff tools for all content types
 - Links, dependencies and process tracing
 - Indexed search for all content types

Manageable programming process

Questions and Answers

