

CDISC Migration

PhUSE 2010 - Berlin

47 of the top 50 biopharmaceutical firms
use Cytel software to design, simulate and analyze their clinical studies.

Source: The Pharm Exec 50—the world's top 50 pharmaceutical companies according to PHARMACEUTICAL EXECUTIVE

Presentation Objectives

- Introductions
- CDISC Migration Strategies
- Benefits of Deploying a Migration Utility
- Questions & Answers

Introductions

Cytel Presenters	Title
Irving Dark	Vice President, Clinical Research Services
Jim Nigrelli	Principal Statistical Programmer
Ajay Sathe	CEO, Cytel India

CDISC Experience

- Converted data from more than 50 trials to SDTM compliant data sets
- Practice leaders with extensive CDISC experience
- Recent FDA Approval using data migrated to CDISC by Cytel
- Licensed WebSDM™ for checking SDTM compliance

Irving Dark, Vice President Clinical Research Services

Jim Nigrelli, Principal Statistical Programmer

CDISC MIGRATION STRATEGIES

Migration Practices – Key components

Gap Analysis
Trial Design
Annotated CRFs
Data Migration
Define.xml
QC
WebSDM™

- Itemization and evaluation of files to support migration activities
- Initiated prior to commencing migration activities
- Document inventory
- Reconcile sample CRFs versus source data
- Comparison of protocol amendments/versions against CRF versions
- Advantages:
 - Proactive approach rather than reactive
 - Clarifies the scope and challenges of migration activities
 - Identifies differences in data collection formats

- Domain that contains a clear representation of the design of a clinical trial
 - Created prior to CRF annotation and data migration
 - Experience needed to create the trial design data set:
 - SDTM Trial Design Concepts
 - Experience with key clinical documents (e.g. Protocol, SAP, Data Sets, CSR)
 - Examples:
 - Standardization across studies
 - Identifying trial ARMS and trial VISITS
 - Planned per protocol versus actual

- The annotated CRF provides the migration team with a clear presentation of SDTM variables
- Completed following the GAP analysis and prior to migration
- Experience needed to create the annotated CRF:
 - Knowledge of the SDTM Implementation Guide and Metadata Submission Guidelines (chapter 4 – Guidelines for Annotating CRFs)
 - Standard data management principles

- Currently using SAS programs to migrate data:
 - Use SAS and Excel
 - Mapping specifications generated (Excel)
 - SAS Macros to assist in standardizing the migration process (e.g. ISO date conversion, SDTM templates, Control Terminology Look-ups)

- Metadata describing the format and content of the data sets
- In-House Utility:
 - SAS Generated
 - Metadata / Code-list driven
 - CRF page links
 - Comments insertion

- QC of Migration of Clinical Data
- Derivation of SDTM variables
- Generation of Define.xml
- Compliance to SDTM standards
- Key Components:
 - Data Accountability
 - Source data checks
 - Define.xml
 - Additional QC checks
 - SDTM Compliance

- Data review tool used to validate the compliance of data sets and define.xml to SDTM standards
- Identifies:
 - True errors
 - Data related errors (e.g. start / end dates)
 - False errors (e.g. until-less test results)

Ajay Sathe, CEO Cytel India

BENEFITS OF DEPLOYING A MIGRATION UTILITY

Automate – what and how

- Metadata approach
- Tasks most amenable to automation:
 - Mapping of variables (mapping specs document)
 - Mapping Code Generation
 - Validating the mapped datasets

Automate – what and how

- Account for all of Source Data
- Load Target Data definition (SDTM or other)
- Create Custom Domains
- Define global terminology
- Define pre-processing logic, e.g., *TRANSPOSE*
- Supply mapping logic
 - Talk about the seven types of mapping logic?
- Push the button!
 - (no, it's not that simple, of course. Verbal explanation)

So, what's new in this approach?

- Target Structure flexible, not restricted to SDTM
- Data Driven approach – the logic resides in data
 - Logic stored in mapping description datasets.
- Self-documenting. Generate mapping specification in parallel
- Meta-programming. Generate SAS® code that does the job
- Transformation types – versatile & comprehensive
 - *Automatic – match name and type*
 - *Data Type conversion*
 - *Rename Variables*
 - *Value Conversions, using Controlled Terminology*
 - *Date Conversions (e.g. ISODATE)*
 - *Concatenation or Parsing/splitting*
 - *Compute – user defined SAS® statements*

The Outputs

- Account for all of Source Data
- Mapping Specifications Document
- SAS® code for Mapping
- Migrated Datasets

Step 1: Project-level settings

MapGenie : Initial Setup

Open Project ☐ New ☒ Existing

Project ID

Select SDTM Version

Source Library Name and Location

Select	Library Name	Library Location
☐	Extract	F:\Demo\Inputs\CY21-002\extract
☐	pre	F:\Demo\Inputs\CY21-002\presdtm

Target Library Name and Location

Format File

Output Location (Common Path)

Output Location (Code)

Output Location (Logs)

Output Location (Listings)

Startup.sas and Source Metadata generated

Select SDTM Version here

Set various folder paths here

Step 2: Dataset level settings

MapGenie : Dataset Level Mapping

Create Custom Domain

Select Common Variable Dataset: PRE.COMVAR

Define Dataset Level Mapping

Source Datasets

PRE.COMVAR
EXTRACT.AE
EXTRACT.DEMO
EXTRACT.VITALS

STUDYID
USUBJID
_ARM
_ARMCD
_INVTAM
_LDOSDT
_FDOSDT
_RFSTD
_RFSTTM
_RFSTDTN

Map

Target Datasets

DM
CO
CM
EX
SU
AE
DS
MH
DV
FG

Mapping definition begins to appear. Notice the "WHERE"

Select	Source Dataset	Target Dataset	WHERE Condition
☐	EXTRACT.AE	AE	
☐	EXTRACT.VITALS	VS	

Remove
Remove All
Add Filter...

Pick a source dataset

Pick a target dataset (in this case SDTM)

Mapping definition begins to appear. Notice the "WHERE"

Step 3 – global Controlled Terminology

MapGenie:Controlled Terminology

☒ Browse file for controlled terminology

F:\Demo\Inputs\CY21-002\Sponsor SDTM CT .xls

Variables to define controlled terminology

Variable Name
Controlled Terminology

Column names from the file

Variable Name
sponsor- CT variable values

☒ Define controlled terminology manually

Domain Variable

Select	Variable	Controlled Terminology
☐	SEX	UN
☐	SEX	U
☐	SEX	M
☐	SEX	F

Load controlled terminology from an Excel file...

... and / or, add it to mapping metadata manually

Step 4 – mapping the Variables

MapGenie : First pass for variable level mapping

☒ Define Automapping

☒ Select All

- ☒ AE
- ☒ DM
- ☒ VS

Auto Mapping Selection

- ☒ Direct Mapping (When Name and Type both match)
- ☒ Date Variables
- ☒ Suggested data type conversion (When only the name matches)
- ☒ Populate SEQ Variable
- ☒ Define DY variables (Study day)

Auto Mapping Results

Select Target Dataset:

Source Dataset	Source Variable	Target Dataset	Target Variable	Transformation	Macro Cal
		AE	AESEQ	Compute_SEQ	%sequencemap(sort_variab
		AE	AESTDY	Compute_DY	%studydaymap(dtc_date = A
EXTRACT.AE	AEBODSYS	AE	AEBODSYS	AUTO	AEBODSYS = AEBODSYS;
EXTRACT.AE	AEDECOD	AE	AEDECOD	AUTO	AEDECOD = AEDECOD;
EXTRACT.AE	AEENDTC	AE	AEENDTC	ISO 8601 Conversion	%iso8601_date (outputvar =
EXTRACT.AE	AEOUT	AE	AEOUT	AUTO	AEOUT = AEOUT;

“Auto” mapping using smart rules

The rules appear here; User override possible

Step 4 – auto-mapping rules

MapGenie : First pass for variable level mapping

☒ Define Automapping

☒ Select All

- ☒ AE
- ☒ DM
- ☒ VS

Auto Mapping Selection

- ☒ Direct Mapping (When Name and Type both match)
- ☒ Date Variables
- ☒ Suggested data type conversion (When only the name matches)
- ☒ Populate SEQ Variable
- ☒ Define DY variables (Study day)

Auto Mapping Results

Select Target Dataset:

Source Dataset	Source Variable	Target Dataset	Target Variable	Transformation	Macro Cal
		AE	AESEQ	Compute_SEQ	%sequencemap(sort_variab
		AE	AESTDY	Compute_DY	%studydaymap(dtc_date = A
EXTRACT.AE	AEBODSYS	AE	AEBODSYS	AUTO	AEBODSYS = AEBODSYS;
EXTRACT.AE	AEDECOD	AE	AEDECOD	AUTO	AEDECOD = AEDECOD;
EXTRACT.AE	AEENDTC	AE	AEENDTC	ISO 8601 Conversion	%iso8601_date (outputvar =
EXTRACT.AE	AEOU	AE	AEOU	AUTO	AEOU = AEOU;

“Auto” mapping using smart rules

The rules appear here; User override possible

Step 5 – mapping the Variables

Select the domain

Corresponding Source & Targets appear

User sets pre-processing and transformation logic

Mappings specs are saved. Self-documenting!

MapGenie : Variable Level Mapping

Define custom domain variables

Project ID: CY21-002

Select Domain: VS

Select source dataset: EXTRACT.VITALS

Select source variables:

Select	Name	Type	Label
☐	STUDYID	char	Study ID
☐	SITEID	char	Center or Site ID
☐	USUBJID	char	Unique Subject ID
☐	SUBJIDN	num	Subject ID

Map

Select target variables:

VSGRPID
VSSPID
VSTESTCD
VSTEST
VSCAT
VSSCAT
VSPOS

SDTM Recommendation: VSGRPID

Label : Group ID
Core : Perm
Role : Identifier
Type : Char

Mapping specifications:

Select	Source Dataset	Source Variable	Target Domain	Target Variable	Pre-Processing	Pre-Processing Value	Transformation
☐	EXTRACT.VITALS	SYSBP	VS	DOMAIN	Transpose-Name	1	Auto
☐	EXTRACT.VITALS	DIABP	VS	VSTESTCD	Transpose-Name	2	Auto
☐	EXTRACT.VITALS	PULSE	VS	VSTESTCD	Transpose-Name	3	Auto
☐	EXTRACT.VITALS	HEIGHT	VS	VSTESTCD	Transpose-Name	4	Auto
☐	EXTRACT.VITALS	WEIGHT	VS	VSTESTCD	Transpose-Name	5	Auto
☐	EXTRACT.VITALS	TEMP	VS	VSTESTCD	Transpose-Name	6	Auto
☐	EXTRACT.VITALS	SYSBP	VS	VSGRPID	Transpose-Value	1	Auto

Note: Please provide extra inputs for transformations- extract, value mapping, concatenate and Drop : Move to SUPP domain using show form button

Save

View mapping specifications

Export

Generate code

Create dataset

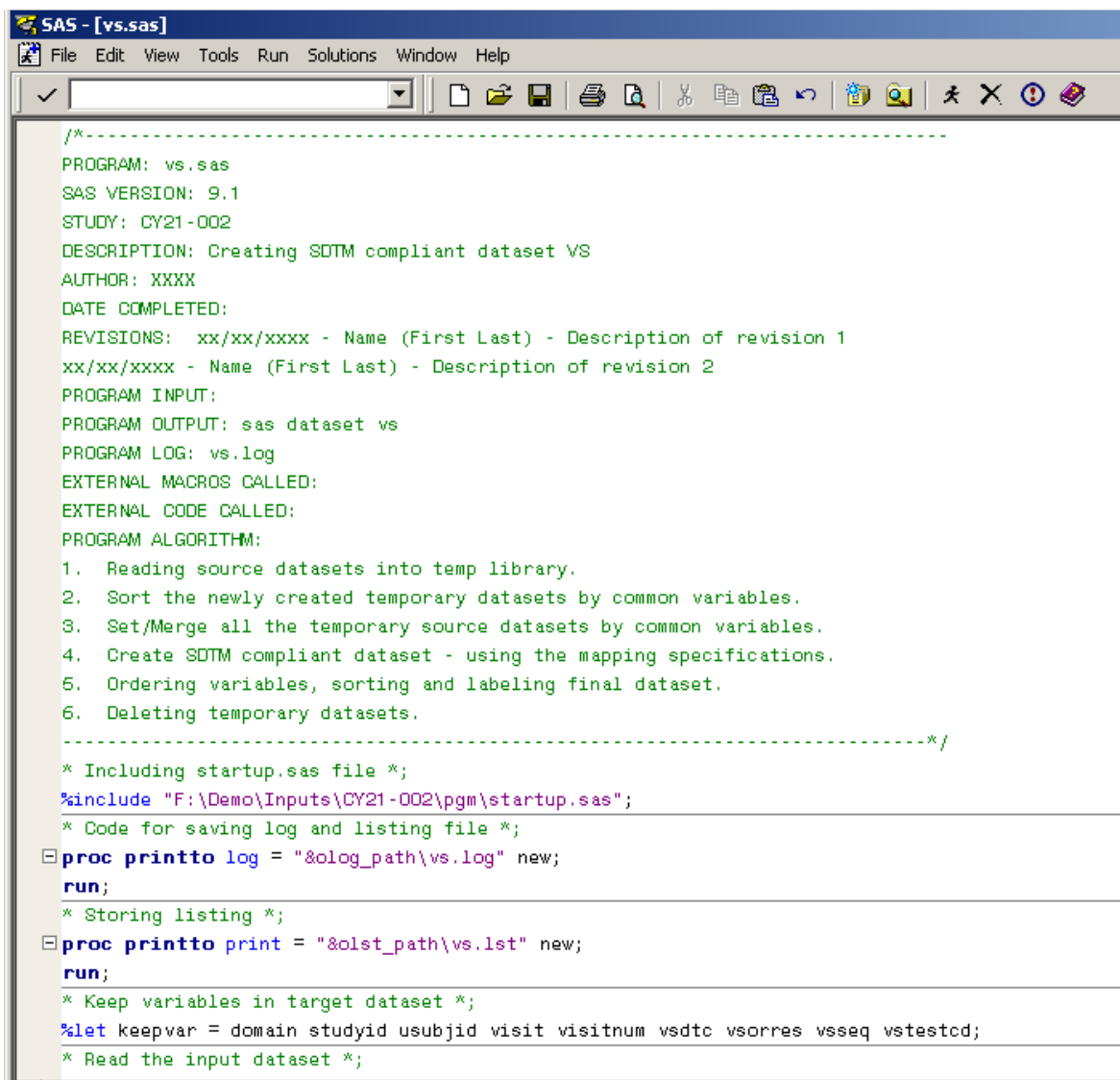
Target and source variables populated

<< Back

Transformation types

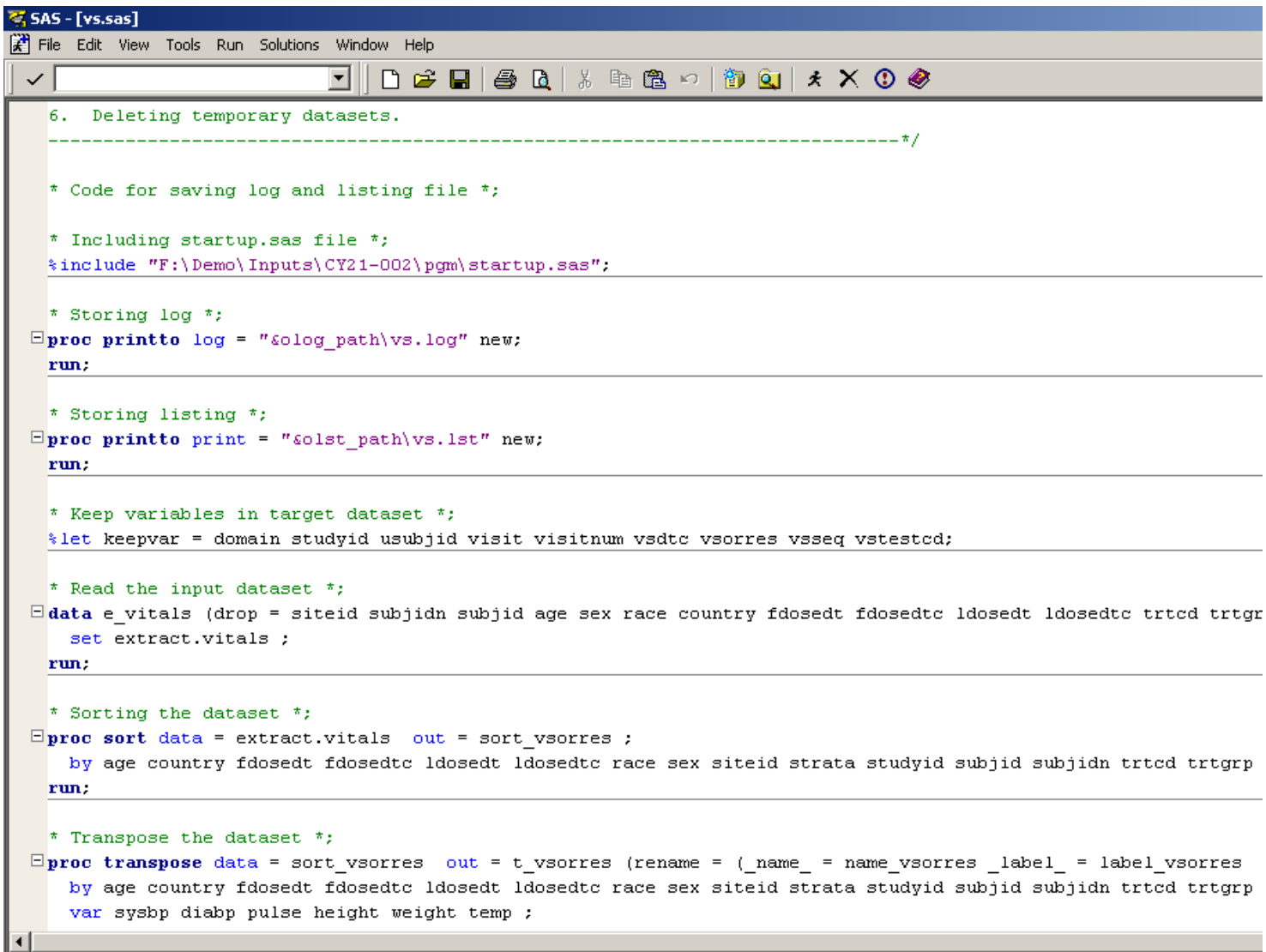
- Auto
- Convert Data Types
- Rename
- Drop
- Convert Values
- Convert ISODate
- Concatenate
- Extract
- Compute (any user-defined SAS® code logic)

The generated code



```
/*-----  
PROGRAM: vs.sas  
SAS VERSION: 9.1  
STUDY: CY21-002  
DESCRIPTION: Creating SDTM compliant dataset VS  
AUTHOR: XXXX  
DATE COMPLETED:  
REVISIONS: xx/xx/xxxx - Name (First Last) - Description of revision 1  
xx/xx/xxxx - Name (First Last) - Description of revision 2  
PROGRAM INPUT:  
PROGRAM OUTPUT: sas dataset vs  
PROGRAM LOG: vs.log  
EXTERNAL MACROS CALLED:  
EXTERNAL CODE CALLED:  
PROGRAM ALGORITHM:  
1. Reading source datasets into temp library.  
2. Sort the newly created temporary datasets by common variables.  
3. Set/Merge all the temporary source datasets by common variables.  
4. Create SDTM compliant dataset - using the mapping specifications.  
5. Ordering variables, sorting and labeling final dataset.  
6. Deleting temporary datasets.  
-----*/  
  
* Including startup.sas file *;  
%include "F:\Demo\Inputs\CY21-002\pgm\startup.sas";  
  
* Code for saving log and listing file *;  
☐ proc printto log = "&olog_path\vs.log" new;  
run;  
  
* Storing listing *;  
☐ proc printto print = "&olst_path\vs.lst" new;  
run;  
  
* Keep variables in target dataset *;  
%let keepvar = domain studyid usubjid visit visitnum vsdtc vsorres vsseq vstestcd;  
  
* Read the input dataset *;
```

The generated code



```
SAS - [vs.sas]
File Edit View Tools Run Solutions Window Help

6. Deleting temporary datasets.
-----*/

* Code for saving log and listing file *;

* Including startup.sas file *;
%include "F:\Demo\Inputs\CY21-002\pgm\startup.sas";

* Storing log *;
proc printto log = "%olog_path\vs.log" new;
run;

* Storing listing *;
proc printto print = "%olst_path\vs.lst" new;
run;

* Keep variables in target dataset *;
%let keepvar = domain studyid usubjid visit visitnum vsdte vsorres vsseq vstestcd;

* Read the input dataset *;
data e_vitals (drop = siteid subjidn subjid age sex race country fdosedt fdosedtc ldosedt ldosedtc trtcd trtgrp
set extract.vitals ;
run;

* Sorting the dataset *;
proc sort data = extract.vitals out = sort_vsorres ;
by age country fdosedt fdosedtc ldosedt ldosedtc race sex siteid strata studyid subjid subjidn trtcd trtgrp
run;

* Transpose the dataset *;
proc transpose data = sort_vsorres out = t_vsorres (rename = (_name_ = name_vsorres _label_ = label_vsorres
by age country fdosedt fdosedtc ldosedt ldosedtc race sex siteid strata studyid subjid subjidn trtcd trtgrp
var sysbp diabp pulse height weight temp ;
```

Thank you.

Questions?

Contact:

Laurent Spiess, Ph.D.

General Manager Europe

laurent@cytel.com

Phone: +33 6 73 86 12 54

