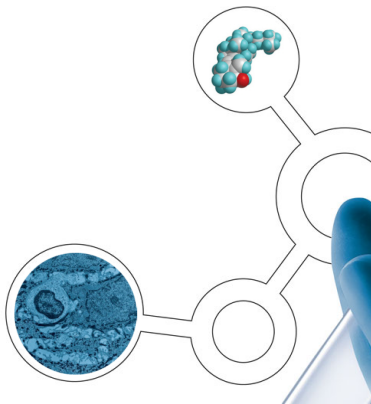
CS05 Creating define.xml from a SAS program

Jørgen Mangor Iversen
LEO Pharma A/S

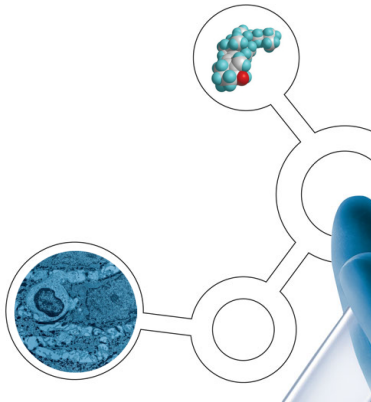
About LEO Pharma A/S

- Independent, research-based pharmaceutical company
- Founded in 1908
- Employs around 5,000 employees worldwide
- Headquartered in Denmark
- Fully owned by the LEO Foundation
- Drugs to dermatologic and thrombotic patients in more than 100 countries
- Vision of becoming the preferred dermatology care partner



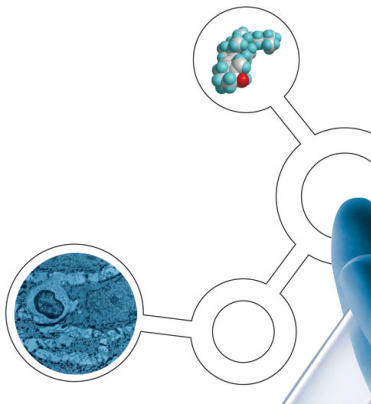
Introduction

- Many ways to produce define.xml
  - Outsource to a CRO or other
  - Specialized XML editors
  - Integrated metadata tools
  - SAS functionality
    - XML Engine
    - XML map
    - Proc Template
    - Data Step and Put statement
- Not a simple choice



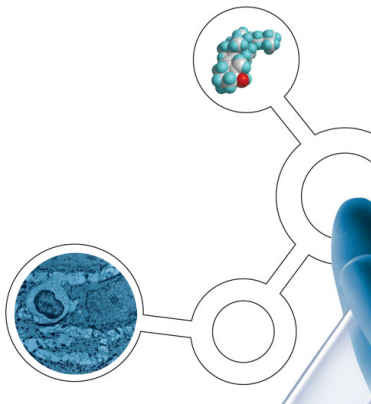
Metadata

- Define.xml is all about metadata, not study data
  - CDISC standards
    - Implementation Guides
    - Download from member area
    - Download from nci.gov
  - SAS datasets
    - Proc Contents
    - Dictionary tables
  - Study specific, collect somehow
    - Excel
    - Database
    - SAS Datalines
    - Web application
    - Handwritten notes
    - Other



Standard Metadata

- Implementation Guideline copy/paste
  - Easy to make a mistake
  - Only way for ADaM
- Download standards
  - SDTM 3.1.2 contains document errors
    - Embedded '0AA0'x hexadecimal values
    - SUPP-- variables with blank roles
  - NCI Controlled Terminology hard to automate
    - First sheet is a readme
    - Terminology sheet contains a date that varies



Standard Metadata Example

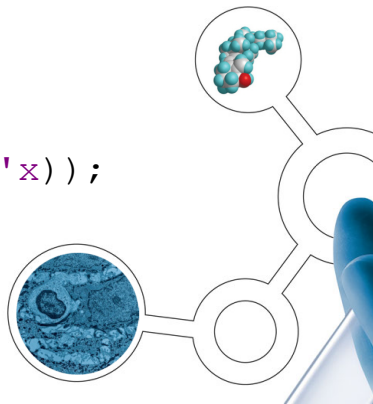
Seq. For Order	Observation Class	Domain Prefix	Variable Name (minus domain prefix)	Variable Name	Variable Label	Type	Controlled Terms or Format	Role	Core
29	All Classes		STTPT	--STTPT	Start Reference Time Point	Char		Timing	
30	All Classes		ENRTPT	--ENRTPT	End Relative to Reference Time Point	Char		Timing	
31	All Classes		ENTPT	--ENTPT	End Reference Time Point	Char		Timing	
1	Special-Purpose	DM	STUDYID	STUDYID	Study Identifier	Char		Identifier	Req
2	Special-Purpose	DM	DOMAIN	DOMAIN	Domain Abbreviation	Char	DM	Identifier	Req
3	Special-Purpose	DM	USUBJID	USUBJID	Unique Subject Identifier	Char		Identifier	Req
4	Special-Purpose	DM	SUBJID	SUBJID	Subject Identifier for the Study	Char		Topic	Req
5	Special-Purpose	DM	RFSTDTC	RFSTDTC	Subject Reference Start Date/Time	Char	ISO 8601	Record Qualifier	Exp
6	Special-Purpose	DM	RFENDTC	RFENDTC	Subject Reference End Date/Time	Char	ISO 8601	Record Qualifier	Exp
7	Special-Purpose	DM	SITEID	SITEID	Study Site Identifier	Char		Record Qualifier	Req
8	Special-Purpose	DM	INVID	INVID	Investigator Identifier	Char		Record Qualifier	Perm
9	Special-Purpose	DM	INVNAM	INVNAM	Investigator Name	Char		Synonym Qualifier	Perm
10	Special-Purpose	DM	BRTHDTC	BRTHDTC	Date/Time of Birth	Char	ISO 8601	Record Qualifier	Perm
11	Special-Purpose	DM	AGE	AGE	Age	Num		Record Qualifier	Exp
12	Special-Purpose	DM	AGEU	AGEU	Age Units	Char	(AGEU)	Variable Qualifier	Exp
13	Special-Purpose	DM	SEX	SEX	Sex	Char	(SEX)	Record Qualifier	Req
14	Special-Purpose	DM	RACE	RACE	Race	Char	(RACE)	Record Qualifier	Exp
15	Special-Purpose	DM	ETHNIC	ETHNIC	Ethnicity	Char	(ETHNIC)	Record Qualifier	Perm
16	Special-Purpose	DM	ARMCD	ARMCD	Planned Arm Code	Char	*	Record Qualifier	Req
17	Special-Purpose	DM	ARM	ARM	Description of Planned Arm	Char	*	Synonym Qualifier	Req
18	Special-Purpose	DM	COUNTRY	COUNTRY	Country	Char	(COUNTRY) ISO 3166	Record Qualifier	Req
19	Special-Purpose	DM	DTC	DMDTC	Date/Time of Collection	Char	ISO 8601	Timing	Perm

Downloading Metadata

```
/* Refer to the spread sheet in the members area of cdsic.org using
username and password of LEO membership */
filename cdisc http
'http://www.cdisc.org/stuff/contentmgr/files/0/1723b0630721bd289647c00f6a8
4e0af/misc/sdtmv1_2_sdtmigv3_1_2_2009_03_27_final.xls'
         user=XXXX pass=XXX;

/* Download the spread sheet to a temporary location */
data _NULL_;
  n= -1;
  infile cdisc recfm=s nbyte=n length=len;
  file   '..\cdisc.xls' recfm=n;
  input;
  put _infile_ $varying32767. len;
run;

/* Convert the downloaded spread sheet to a SAS dataset */
proc import datafile='..\cdisc.xls'
                 out=cdisc
               dbms=excelcs
               replace;
run;
```

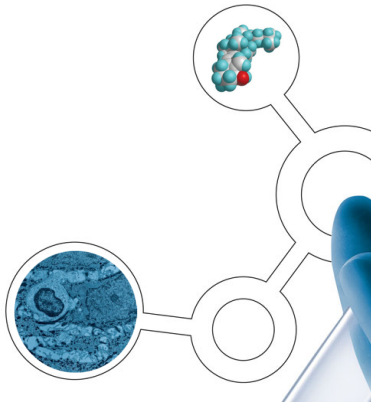
Fixing Metadata

```
/* Fix known bugs in CDSIC spread sheet */
proc sql;
  update cdisc
     set role = 'Record Qualifier'
   where role = ''
     and substr(variable_name, 1, 5) = 'IDVAR';

  update cdisc
     set role = 'Identifier'
   where role = ''
     and variable_name in ('QNAM' 'STUDYID' 'USUBJID' 'RELID' 'RDOMAIN');

  update cdisc
     set role = 'Grouping Qualifier'
   where role = ''
     and variable_name in ('QEVAL' 'QORIG' 'RELTYPE' 'QLABEL');

  update cdisc
     set role = 'Variable Qualifier'
   where role = ''
     and variable_name = 'QVAL';

  update cdisc
     set Controlled_Terms_or_Format =
         strip(translate(Controlled_Terms_or_Format, ' ', '0AA0'x));
quit;
```



Restructuring Standard Metadata

- Create a superset of all possible datasets and variables
  - --SUPP described once, needed for all
    - Except SUPPQUAL and RELREC
  - Add generic variable to all domains
    - Merge by Class attribute
- Transform to hierarchy
  - Global
  - Dataset
  - Variable
  - Value
  - Computational Methods



Create --SUPP Domains

```
/* Create one SUPP-- domain for each CDISC domain, except SUPPQUAL and RELREC */
create table supp_datasets as select distinct
       'SUPP' || dataset                                                as Dataset,
       'Supplemental Qualifiers for ' || dataset                        as Description,
       'One record per IDVAR, IDVARVAL, and QNAM value per subject'     as Structure,
       'Relationship'                                                   as Class,
       'Tabulation'                                                     as Purpose,
       'STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM'               as Keys
  from cdisc_datasets
 where dataset not in ('SUPPQUAL' 'RELREC')
 order by dataset;

/* Add the SUPP-- domains to the rest of the domains */
create table datasets as select distinct *
  from cdisc_datasets
 union select distinct *
  from supp_datasets
 order by dataset;
```

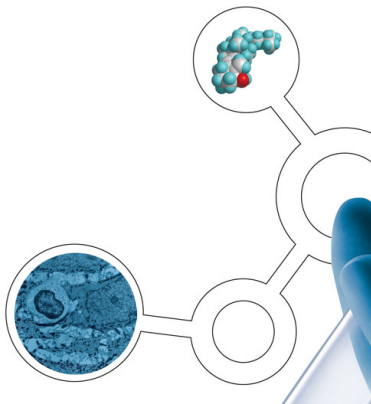
Controlled Terminology Example

Code	Codelist Code	Codelist Extensible (Yes/No)	Codelist Name	CDISC Submission Value	CDISC Synonym(s)
C66767		No	Action Taken with Study Treatment	ACN	Action Taken with Study Treatment
C49503	C66767		Action Taken with Study Treatment	DOSE INCREASED	
C49504	C66767		Action Taken with Study Treatment	DOSE NOT CHANGED	
C49505	C66767		Action Taken with Study Treatment	DOSE REDUCED	
C49501	C66767		Action Taken with Study Treatment	DRUG INTERRUPTED	
C49502	C66767		Action Taken with Study Treatment	DRUG WITHDRAWN	
C48660	C66767		Action Taken with Study Treatment	NOT APPLICABLE	NA; Not Applicable
C17998	C66767		Action Taken with Study Treatment	UNKNOWN	U; Unknown
C66768		No	Outcome of Event	OUT	Outcome of Event
C48275	C66768		Outcome of Event	FATAL	Grade 5; 5; FATAL
C49494	C66768		Outcome of Event	NOT RECOVERED/NOT	
C49498	C66768		Outcome of Event	RECOVERED/RESOLV	
C49495	C66768		Outcome of Event	RECOVERED/RESOLV	
C49496	C66768		Outcome of Event	RECOVERING/RESOLV	
C17998	C66768		Outcome of Event	UNKNOWN	U; Unknown

README | SDTM Terminology 2013-04-12

Libname to Excel Spread Sheet

```
/* Get full path of 'nci.xls' */
filename nci 'nci.xls';
proc sql noprint;
  select length(xpath)
    into :length
    from dictionary.extfiles
   where xpath ? 'nci.xls';

  select strip(xpath) length=&length
    into :xpath
    from dictionary.extfiles
   where xpath ? 'nci.xls';
quit;

/* Refer to the spread sheet locally */
libname nci pcfiles path="&xpath.";
```

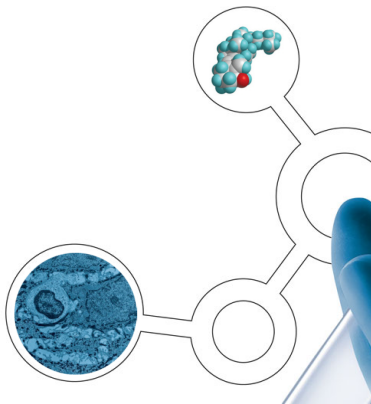


Import a named Excel Sheet

```
/* Get the name of the sheet (contains a date) */
proc sql noprint;
  select memname
    into :nci
    from dictionary.tables
   where upcase(libname) = 'NCI'
     and upcase(memname) not contains 'README';
quit;

/* Convert the downloaded spread sheet to a SAS dataset */
proc import datafile="&xpath."
                 out=nci
                dbms=excelcs
                replace;
  sheet="&nci.";
run;
```

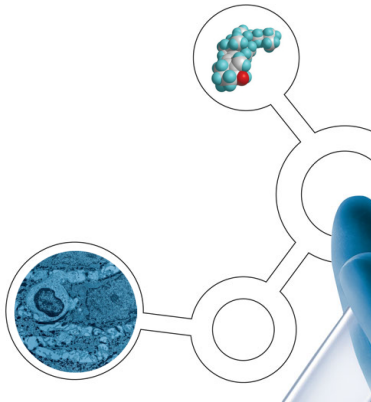
Restructuring Controlled Terminology

- Code list names and values on separate lines
- Extract names and values separately
- Merge by Code = Codelist Code



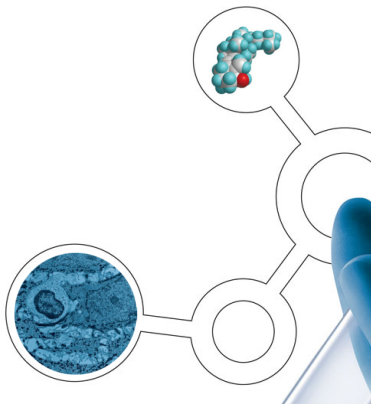
Extract Components

```
/* Metadata for each code list */
create table lists as select distinct
       codelist_name           as Codelist_Name,
       cdisc_submission_value as Codelist,
       codelist_extensible__yes_no_ as Extensible
  from nci
 where codelist_code = ''
   and codelist_name ne '';

/* Metadata for each code list value */
create table names as select distinct
       codelist_name           as Codelist_Name,
       cdisc_submission_value as Code_Value,
       NCI_preferred_term      as Decode_Value
  from nci
 where codelist_code ne '';
```



Merge Controlled Terminology

```
/* All metadata in all rows */
create table Codelist as select distinct
       Codelist,
       lists.Codelist_Name,
       Code_Value,
       Decode_Value,
       '' as Extern_Dict_Ver,
       Extensible
  from lists,
       names
 where lists.Codelist_Name = names.Codelist_Name
 order by Codelist, Codelist_Name;
```



Global Metadata

#	Variable	Type	Length	Format	Informat	Label
1	ODMVersion	Character	8			
2	DefineVersion	Character	8			
3	SDTMName	Character	16			
4	SDTMVersion	Character	8			
5	ADaMName	Character	16			
6	ADaMVersion	Character	8			

	ODMVersion	DefineVersion	SDTMName	SDTMVersion	ADaMName	ADaMVersion
▶ 1	1.2	1.0.0	CDISC SDTM	3.1.2	CDISC ADaM	2.1

Dataset Level Metadata

#	Variable	Type	Length	Format	Informat	Label
1	Standard	Character	4			Standard
2	Dataset	Character	32			Dataset
3	Description	Character	100			Description
4	Class	Character	32			Class
5	Structure	Character	200			Structure
6	Purpose	Character	200			Purpose
7	Keys	Character	200			Keys
8	Location	Character	200			Location

	Standard	Dataset	Description	Class	Structure	Purpose	Keys	Location
62	SDTM	SUPPVS	Supplemental Qualifiers for VS	Special-Purpose Datasets	One record per IDVAR, IDV...	Tabulation	STUDYID, RDOMAIN,...	suppvs.xpt
63	SDTM	X*	Please supply domain label	Please supply class	Please supply structure	Please supply purpose	Please supply keys	x*.xpt
64	SDTM	Y*	Please supply domain label	Please supply class	Please supply structure	Please supply purpose	Please supply keys	y*.xpt
65	SDTM	Z*	Please supply domain label	Please supply class	Please supply structure	Please supply purpose	Please supply keys	z*.xpt
66	ADAM	ADSL	Subject Level Analysis	ADSL	One record per subject	Analysis	STUDYID, USUBJID	adsl.xpt
67	ADAM	ADAE	Adverse Events Analysis	Other	One record per adverse eve...	Analysis	STUDYID, USUBJID, ...	adae.xpt
68	ADAM	ADCM	Concomitant Medications Analysis	Other	One record per event per su...	Analysis	STUDYID, USUBJID, ...	adcm.xpt
69	ADAM	ADMH	Medical History Analysis	Other	One record per event per su...	Analysis	STUDYID, USUBJID, ...	admh.xpt
70	ADAM	ADEX	Exposure Analysis	BDS	One record per analysis time...	Analysis	STUDYID, USUBJID, ...	adex.xpt

Variable Level Metadata

#	Variable	Type	Length	Format	Informat	Label
1	Standard	Character	4			
2	Dataset	Character	32			
3	Variable	Character	32			
4	Label	Character	100			
5	DataType	Character	8			
6	ControlledTerminology	Character	200			
7	Role	Character	200			
8	Core	Character	24			
9	Comment	Character	1000			
10	Order	Numeric	8			
11	Class	Character	22			

	Standard	Dataset	Variable	Label	DataType	ControlledTerminology	Role	Core	Comment	Order	Class
7454	SDTM	DM	STUDYID	Study Identifier	Char		Identifier	Req		1	
7455	SDTM	DM	DOMAIN	Domain Abbreviation	Char	DM	Identifier	Req		2	
7456	SDTM	DM	USUBJID	Unique Subject Identifier	Char		Identifier	Req		3	
7457	SDTM	DM	DMSEQ	Sequence Number	Num		Identifier	Perm		4	
7458	SDTM	DM	SUBJID	Subject Identifier for the Study	Char		Topic	Req		4	
7459	SDTM	DM	DMGRPID	Group ID	Char		Identifier	Perm		5	
7460	SDTM	DM	RFSTDTC	Subject Reference Start Date/Time	Char	ISO 8601	Record Qualifier	Exp		5	
7461	SDTM	DM	DMREFID	Reference ID	Char		Identifier	Perm		6	
7462	SDTM	DM	RFENDTC	Subject Reference End Date/Time	Char	ISO 8601	Record Qualifier	Exp		6	

Value Level Metadata

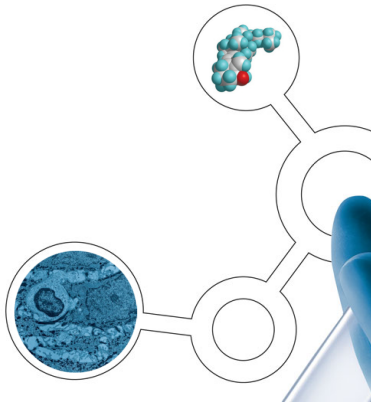
#	Variable	Type	Length	Format	Informat	Label
1	Standard	Character	4			
2	Reference	Character	64			
3	Code	Character	80			
4	Text	Character	80			

	Standard	Reference	Code	Text
1	SDTM	ACN	DOSE INCREASED	Dose Increased
2	SDTM	ACN	DOSE NOT CHANGED	Dose Not Changed
3	SDTM	ACN	DOSE REDUCED	Dose Reduced
4	SDTM	ACN	DRUG INTERRUPTED	Drug Interrupted
5	SDTM	ACN	DRUG WITHDRAWN	Drug Withdrawn
6	SDTM	ACN	NOT APPLICABLE	Not Applicable
7	SDTM	ACN	UNKNOWN	Unknown
8	SDTM	OUT	FATAL	Death Related to Adverse Event
9	SDTM	OUT	NOT RECOVERED/NOT RESOLVED	Not Recovered or Not Resolved
10	SDTM	OUT	RECOVERED/RESOLVED	Recovered or Resolved
11	SDTM	OUT	RECOVERED/RESOLVED WITH SEQUELAE	Recovered or Resolved with Sequelae
12	SDTM	OUT	RECOVERING/RESOLVING	Recovering or Resolving
13	SDTM	OUT	UNKNOWN	Unknown
14	SDTM	AGESPAN	ADOLESCENT (12-17 YEARS)	Adolescent
15	SDTM	AGESPAN	ADULT (18-65)	Adult 18-65 Years Old

Computational Methods Metadata

#	Variable	Type	Length	Format	Informat	Label
1	Identifier	Character	32			
2	Method	Character	200			

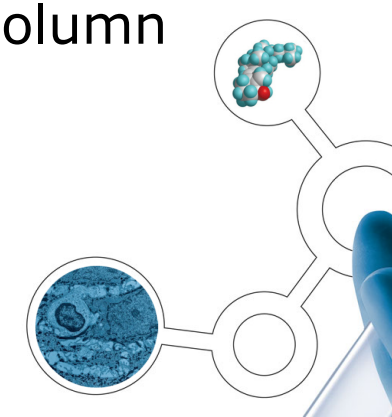
	Identifier	Method
1	COMPMETHOD.PVALUE	P-Values are formatted as: low-<0.001 = '<0.001', 0.001-0.1 = '[5.3]', 0.1-1 = '[4.2]'
2	COMPMETHOD.DURATION	Duration in days, calulated from the dates only, are calculated as: ENDDATE - STARTDATE + 1



Study Specific Metadata

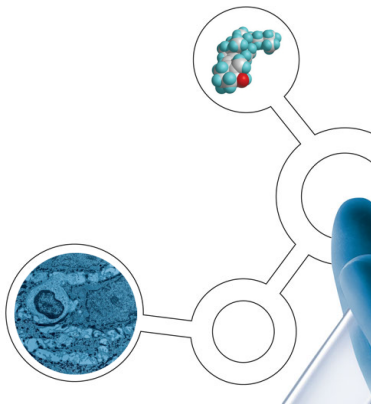
	A	B	C	D
1	Dataset	Variable	Item	Text
2			STANDARD	SDTM
3			FOLDER	MBL 0801 INT
4			STUDY	MBL 0801 INT
5			PROTOCOL	MBL 0801 INT
6			DESCRIPTION	MBL 0801 INT
7			MEDDRAVERSION	13.1
8			ANNOTATED	blank-crf.pdf
9			PATH	
10			ANNOTITLE	Blank CRF
11		STUDYID	Controlled Terms or Format	LEO 80185-MBL 0801 INT
12		RDOMAIN	COMMENT	Any data dependent of USUBJID only, refers to the DM domain
13		STUDYID	ORIGIN	Derived
14		STUDYID	COMMENT	Study code from protocol
15		DOMAIN	ORIGIN	Derived
16		USUBJID	ORIGIN	Derived
17		USUBJID	COMMENT	USUBJID='STUDYID_SUBJID_SITEID'
18	FA		KEYS	STUDYID, USUBJID, FATESTCD, FAOBJ, VISITNUM
19	LB		KEYS	STUDYID, USUBJID, LBTESTCD, VISITNUM
20	QS		KEYS	STUDYID, USUBJID, QSCAT, QSTESTCD, VISITNUM
21	VS		KEYS	STUDYID, USUBJID, VSTESTCD, VISITNUM
22	AE	AESEQ	ORIGIN	Derived
23	AE	AETERM	COMMENT	Also see blank-crf.pdf page 25

- Spread sheet filled by Study Statistician
- Address a ‘cell’ in define.xml by dataset, variable, column
- Blanks refer across filled values



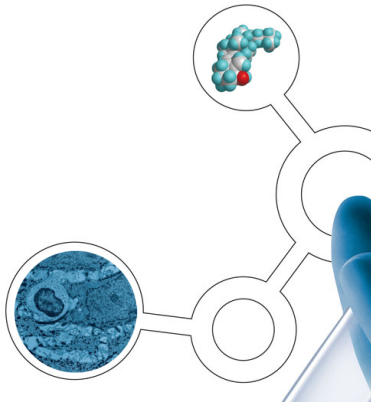
Define.xml structure

- One more level than the metadata
  - Global level data concerning the entire study
  - Dataset level data
  - Variable level data
  - Computational Methods
  - Value level data
  - Controlled Terminology data
- Distinction between value level and Controlled Terminology
- Two standards for the same thing



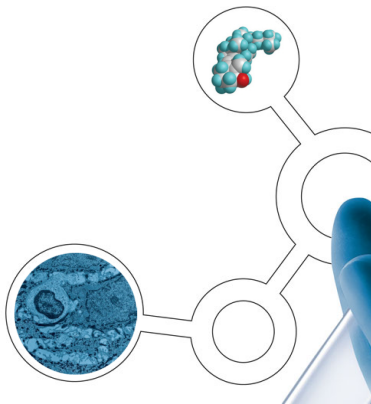
Reconciling with Study Data

- Pruning the metadata
- Strip off parts of metadata not represented in study data
  - Domain datasets not present
  - Variables not used
  - Values of Controlled Terminology not used
  - Values are collected from study data



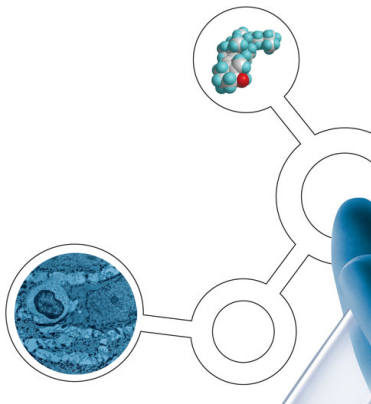
Transformations

- Assigning a category for each dataset
- Add study specific metadata to datasets
- Create metadata for any custom domains
- Resolve class discrepancies between variables and datasets
- Detailing data types (dates, floating points)
- Register user supplied metadata (ORIGIN and COMMENT)
- Add default origins
  - For the LB domain, the default origin is 'Laboratory‘
  - All other SDTM domains have default origin 'CRF’



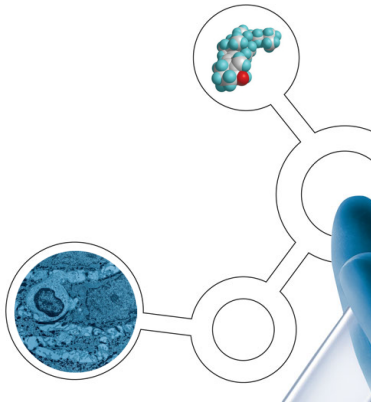
Add User Metadata

```
update study_datasets a
   set description = (select text
                        from study_meta b
                       where a.standard = b.standard
                         and a.dataset  = b.dataset
                         and b.variable = ''
                         and item       =
'DESCRIPTION')
 where exists (select *
                 from study_meta c
                where a.standard = c.standard
                  and a.dataset  = c.dataset
                  and item       = 'DESCRIPTION');
```



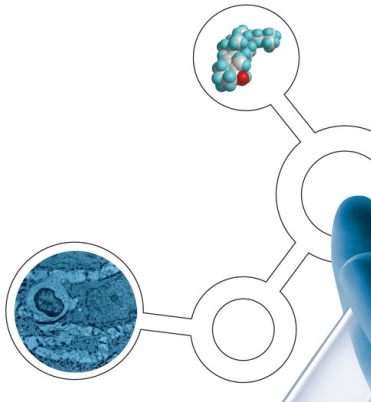
Controlled Terminology

- A constant, possibly several words
  - I.e. a domain name, ISO standard, list of values
- An asterisk (*)
  - Extract values from study data
- An identifier in parenthesis
  - Limit Controlled Terminology to values in the study data
- --BODSYS and --DECOD are special cases
  - Refer to MedDRA Adverse Events Dictionary as external code list



Asterisk (*) Controlled Terminology

- Select values for each variable
- Function htmlencode(var, "amp gt lt apos quot 7bit") ensures data values are XML compliant
- Blank values allowed when other values exist
- Variables with only one value are changed to constant Controlled Terminology



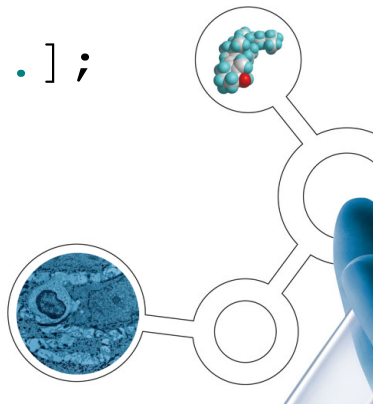
Select Values per Variable

```
/* Collect values from actual data */
data _NULL_;
  set ctfreq;
  where variable ne 'CMDECOD';
  call execute('proc sql;create table values as select distinct');
  call execute('"' || trim(dataset)  || '" as dataset  length=32,' );
  call execute('"' || trim(variable) || '" as variable length=32,');
  select (datatype);
    when ('text')
      call execute('htmlencode('  || trim(variable) || ', "amp gt lt apos quot 7bit") as value length=200');
    when ('integer')
      call execute('left(put('    ||      variable  || ', 32.))                      as value length=200');
    when ('float')
      call execute('left(put('    ||      variable  || ', 32.16))                    as value length=200');
    when ('date')
      call execute('left(put('    ||      variable  || ', ' || trim(format) || '))   as value length=200');
    otherwise;
  end;
  call execute(',count(distinct ' ||      variable  || ')                            as count');
  call execute('from ' || "&standard.." || trim(dataset) || ';quit;');
  call execute('proc append base=discretelist data=values force;run;');
run;
```

Dependant Controlled Terminology

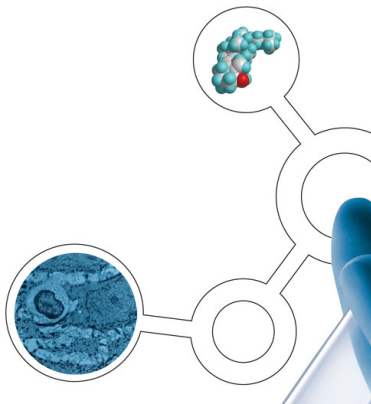
- --ORRES pointed to by a --TESTCD value
- AVAL/AVALC pointed to by a PARAMCD value
- For each value of --TESTCD/PARAMCD
  - Determine data type of --ORRES/AVAL/AVALC
  - Dates identified by **anydate.** format
  - Integers identified by **best.** format
  - Floating point identified by decimal point in --ORRES/AVALC value

```
proc format;
   invalue anydate
   19000101-99993112 = [yymmdd8.]
   other             = [anydtdte8.];
run;
```



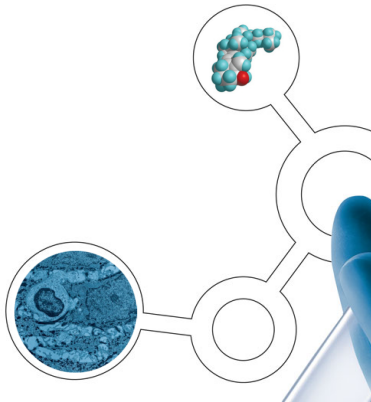
Controlled Terminology in Parenthesis

- When identifier in parenthesis is a known reference
- Select values for each variable
  - Collect only values used
  - Collect additional values for expandable code lists



Creating define.xml

- Simplest possible method
- Pretty print of metadata
- Independent of SAS version
- Data _NULL_;
- PUT statements
- Initial **FILE &xmlfile.;**
- Subsequent **FILE &xmlfile. MOD;**
- Definitions referred to

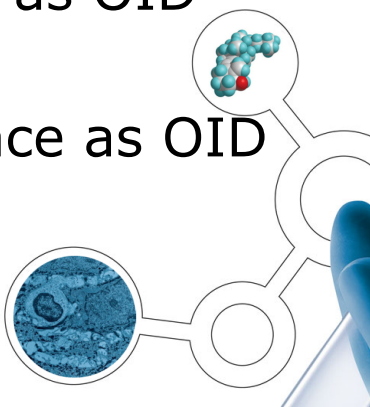


Writing define.xml

```
/* ValueList section */
data _NULL_;
  set    valuetypes;
  where  controlledterminology = '';
  by     dataset variable notsorted;
  retain ordernumber 0;
  file   &xmlfile. mod;

  if first.variable then do;
    ordernumber = 0;
    put '<def:ValueListDef OID="ValueList.' dataset +(-1) '.' variable +(-1) '">';
  end;

  ordernumber = ordernumber + 1;
  put '<ItemRef ItemOID="' dataset +(-1) '.' variable +(-1) '.' value +(-1)
      '" OrderNumber="' ordernumber +(-1) '" Mandatory="No"/>';

  if last.variable then
    put '</def:ValueListDef>';
run;
```

Define.xml Steps

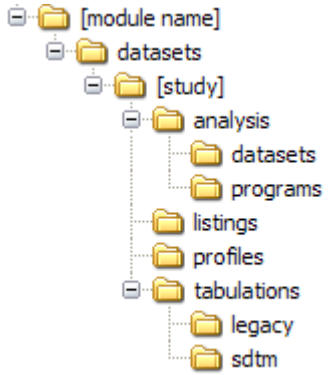
- Global standard, study specific metadata, housekeeping
- def:ComputationMethod, code parts as OID
- def:ValueListDef, datasets and variables as ItemOID
- For each dataset an ItemGroupDef, dataset as OID
  - For each variable an ItemRef, datasets and variable as ItemOID
- Variable details an ItemDef, dataset and the variable as OID
  - Add computational methods, value lists and Controlled Terminology
- ValueList ItemDefs, dataset, variable and value as OID
- Study values in CodeList section, dataset and variable as OID
  - CodeListItem lines, values as CodedValue.
- External code lists as CodeList, dataset and variable as OID
  - A constant text referring to MedDRA version
- Standard controlled terminology in CodeList, reference as OID
  - CodeListItem, codes as CodedValue.



Define.xml

Datasets for Study G21					
Dataset	**Description**	**Structure**	**Purpose**	**Keys**	**Location**
AE	Adverse Events	Events - One record per location per adverse event per subject	Tabulation	STUDYID, USUBJID, AEDECOD, AESTDTC, AELOC	ae.xpt
CM	Concomitant Medications	Interventions - One record per recorded medication occurrence or constant-dosing interval per subject.	Tabulation	STUDYID, USUBJID, CMTRT, CMSTDTC	cm.xpt
CO	Comments	Special Purpose Domains - One record per comment per subject	Tabulation	STUDYID, USUBJID, COSEQ	co.xpt
DA	Drug Accountability	Findings - One record per drug accountability finding per subject	Tabulation	STUDYID, USUBJID, DATESTCD, DADTC	da.xpt
DM	Demographics	Special Purpose Domains - One record per subject	Tabulation	STUDYID, USUBJID	dm.xpt

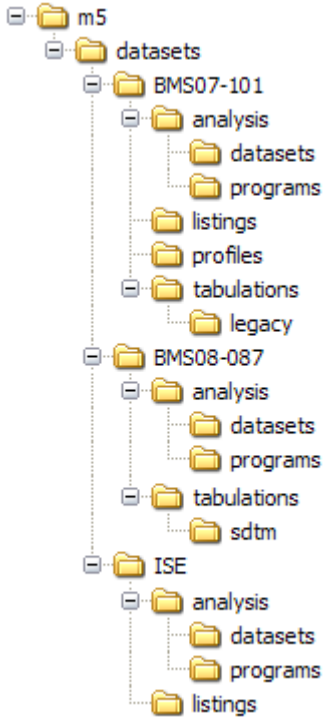
eCTD m5



- LEO Pharma A/S does not submit code
- Everything bundled in a single folder
  - Data
  - Metadata
  - Documentation
- References to XPT and blank-crf.pdf in same folder
- Submit only a document once
- The document *define1-0-0.xsl* is in each define.xml folder
  - Style sheet documents are allowed to be submitted in duplicates

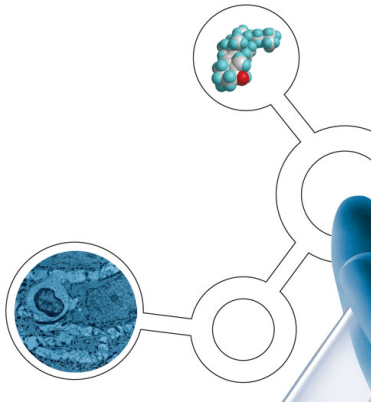
Multiple blank-crf.pdf

- Again, submit only a document once
- Submitting ADaM + SDTM referring same blank-crf.pdf
- Create a link in the ADaM define.xml, referring to the blank-crf.pdf in the folder of the SDTM data
- ..\..\tabulations\sdtm\blank-crf.pdf
  - Relative path



Summary

- No way of showing thousands of code lines
- Highlights
- 750+ lines of code for metadata
  - Split over several program do download and transform
- 1380 lines of code to produce define.xml
  - One big program
  - Restructuring metadata
  - The bulk is reconciling metadata and study data
  - Producing define.xml is minor, once metadata is sorted out



Questions

?