SDTM at Lundbeck

A programmer-driven approach to flexible SDTM creation



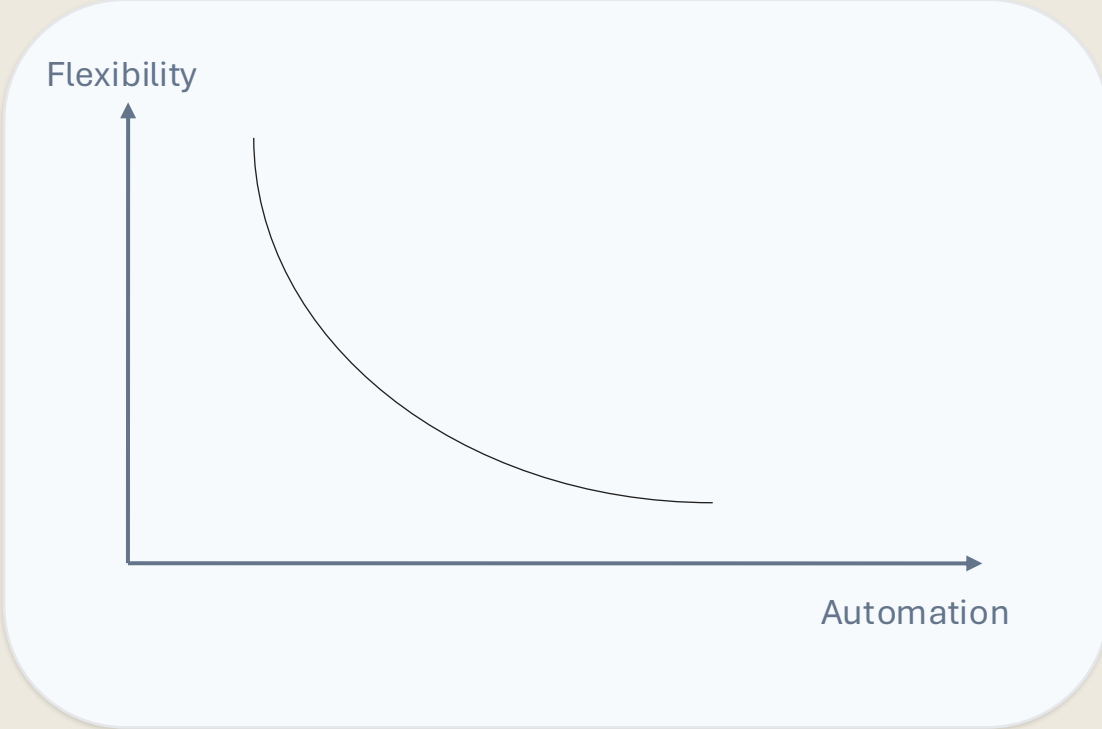
PHUSE event | Mathias Thrane, Lundbeck

What this talk is really about

What this talk is really about

SDTM “system” development with dual focus

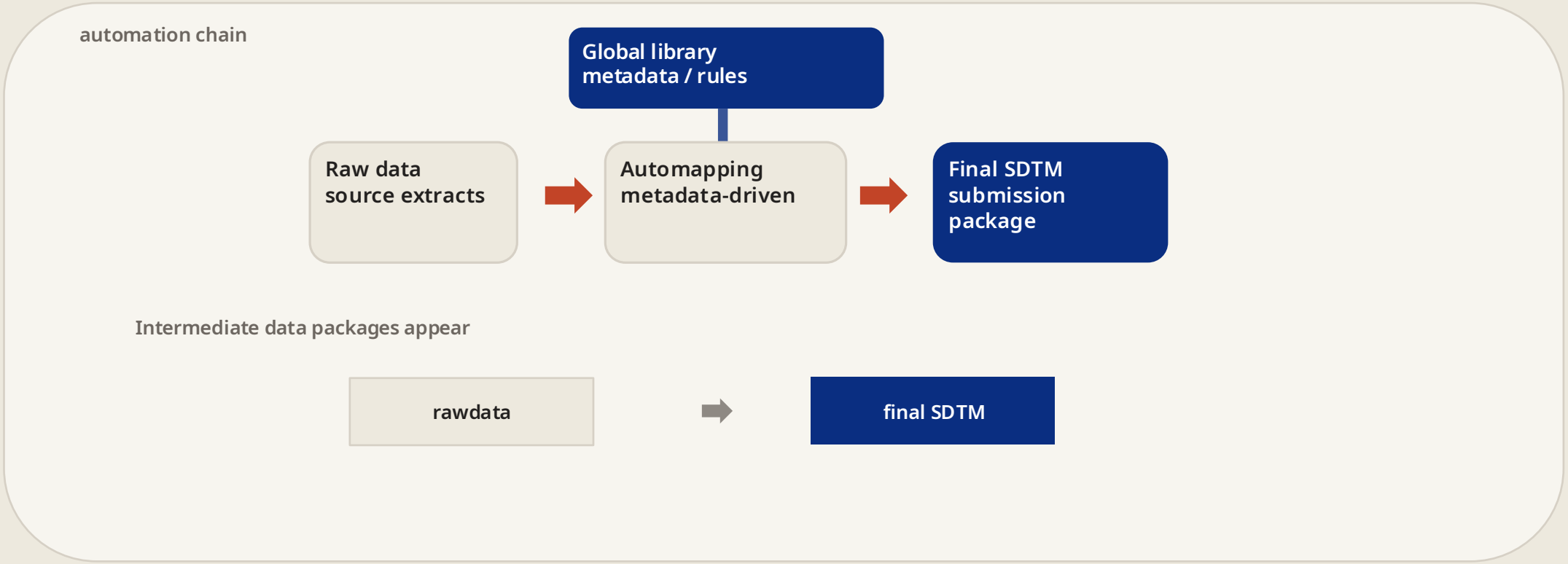
1. Design choices, pro’s n cons
2. How it actually works, hands-on



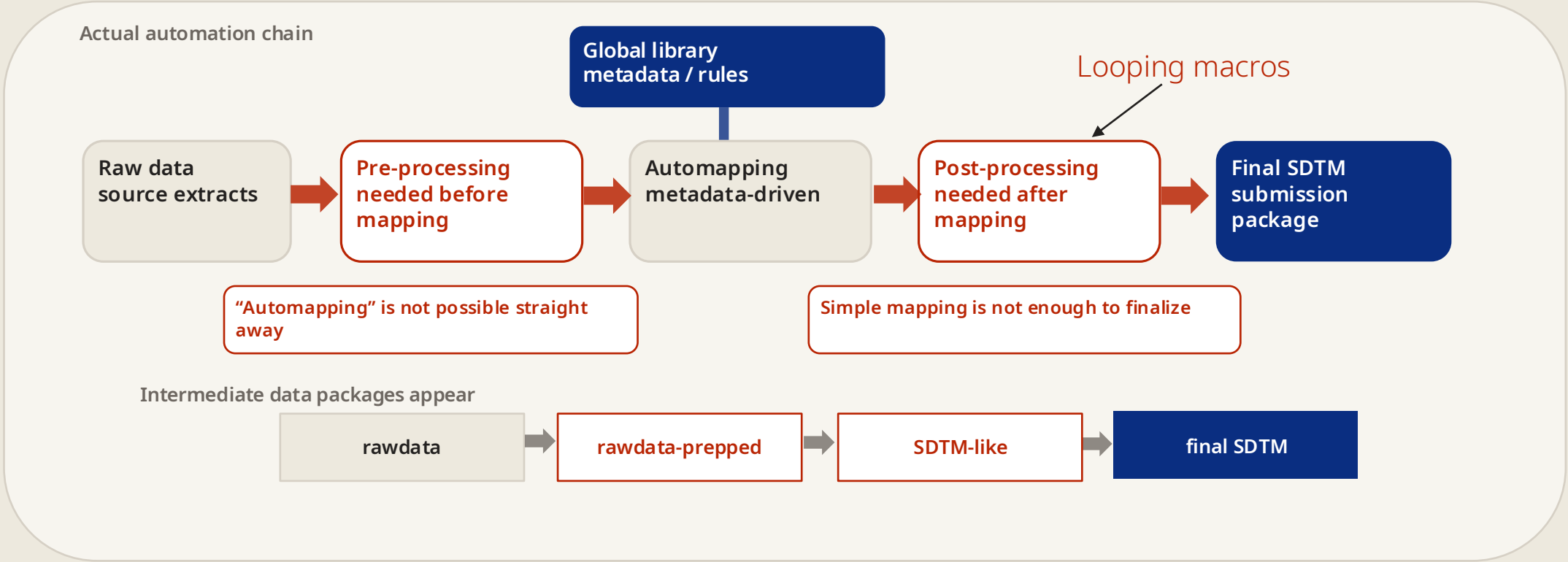
The common SDTM “factory” pattern

‘End-to-end automation’, mapping tool, metadata/global library, standards team

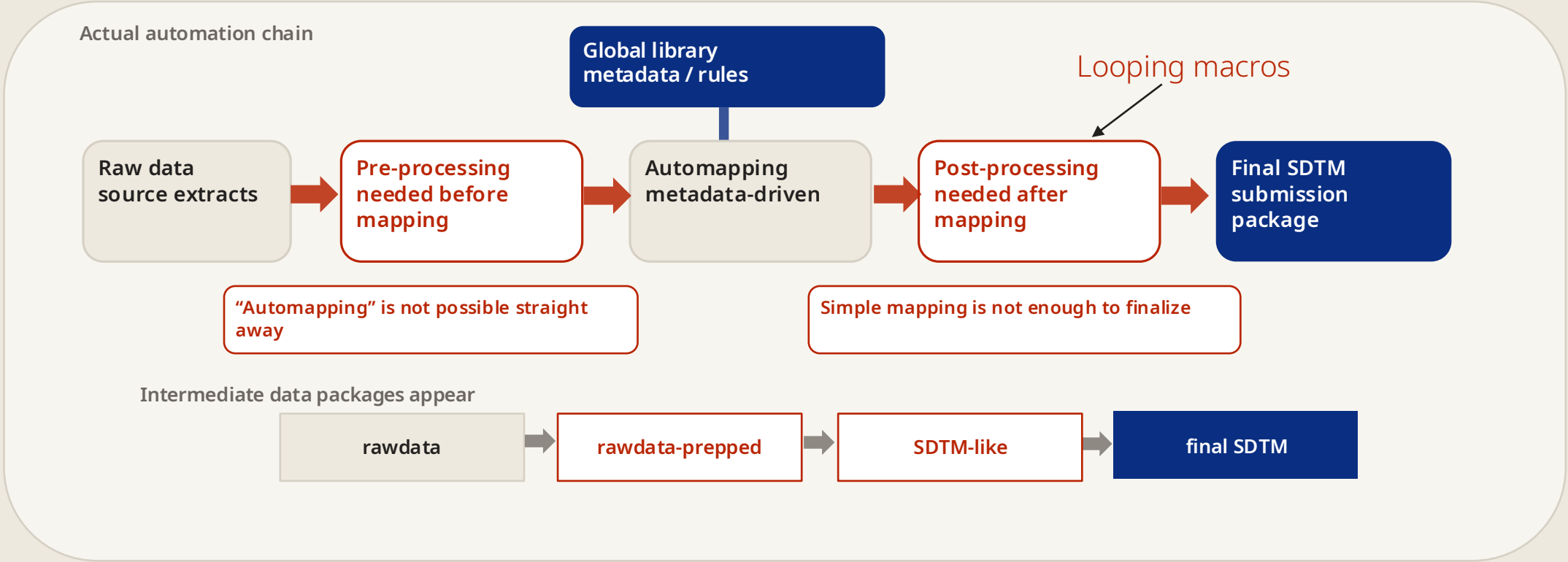
The common SDTM “factory” pattern



The common SDTM “factory” pattern



The common SDTM “factory” pattern



This is not “wrong”. It is optimized for a different balance.

What we did – Design Principles

FLEXIBILITY

Flexibility over automation

Make it easy to handle study-specific complexity.

SAS end-to-end

Homemade standard tools; no paid mapping software dependency.

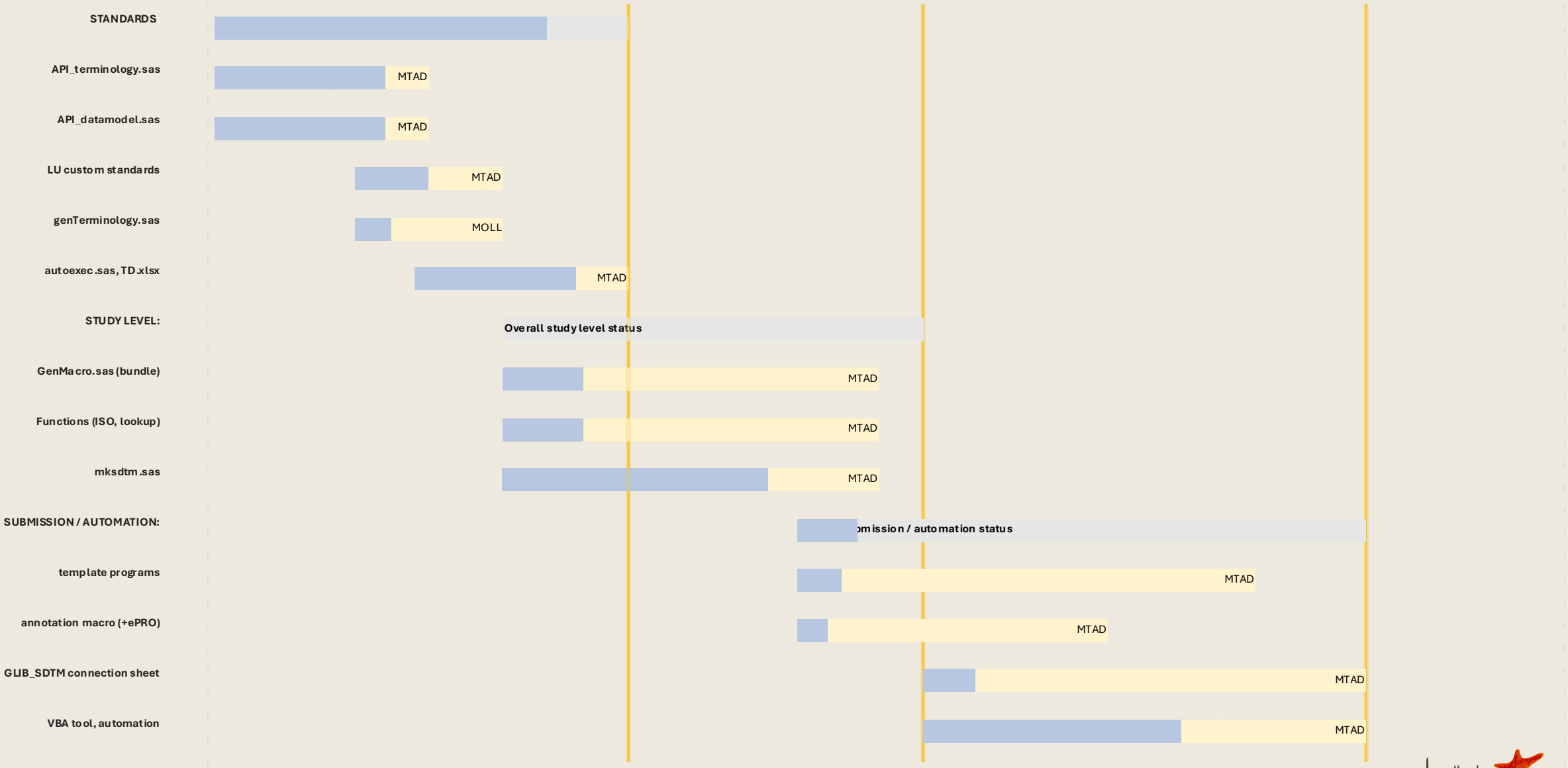
Programming over mapping tools

Transformation logic lives in code, close to the programmer.

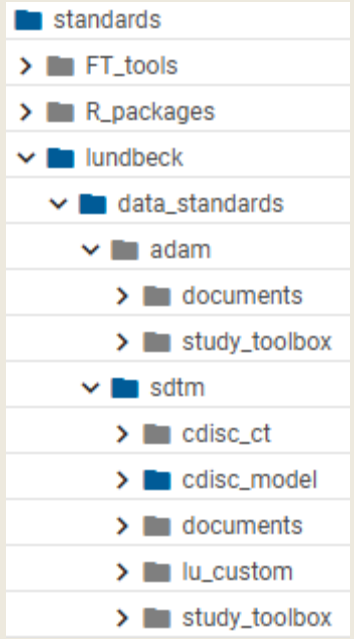
One program → one dataset

A simple mental model with transparent ownership of each output.

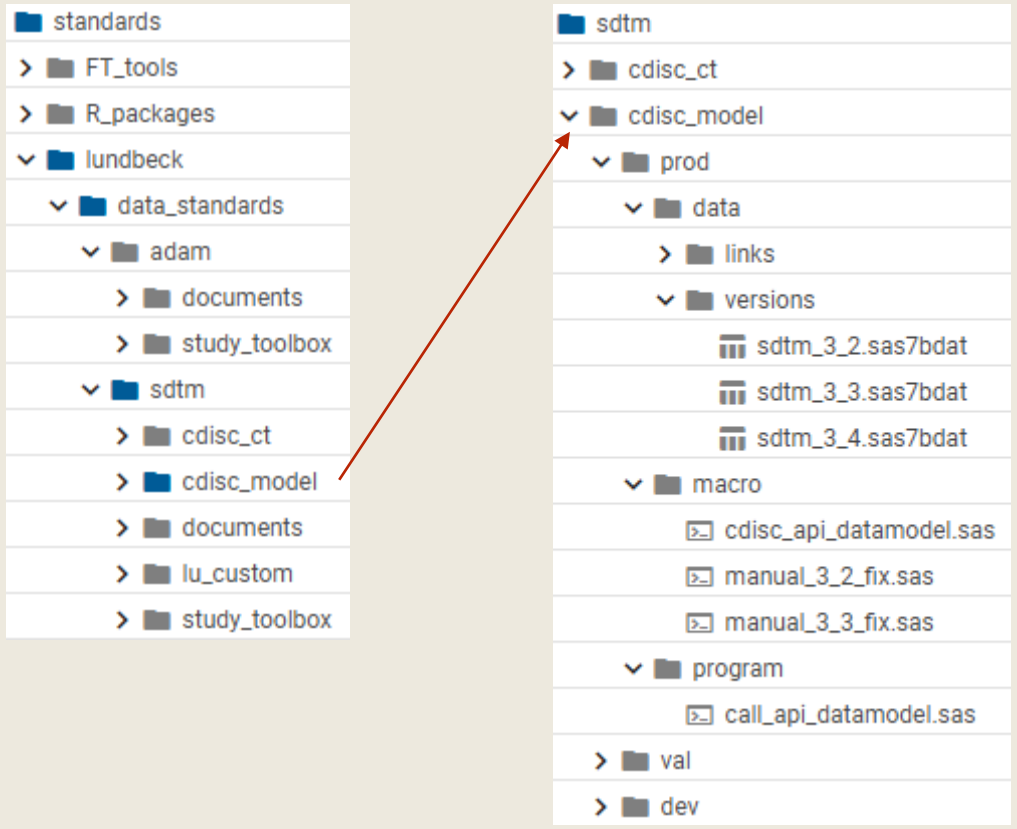
SDTM at Lundbeck



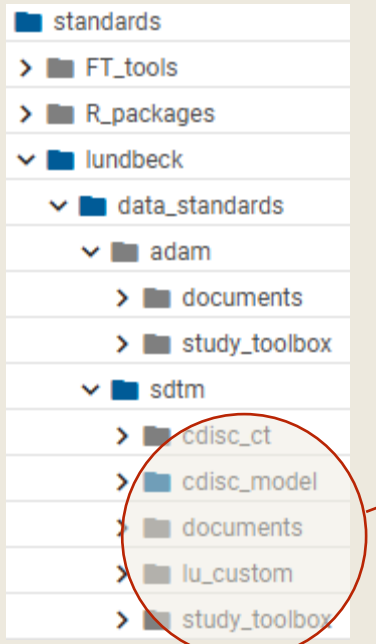
Standards area

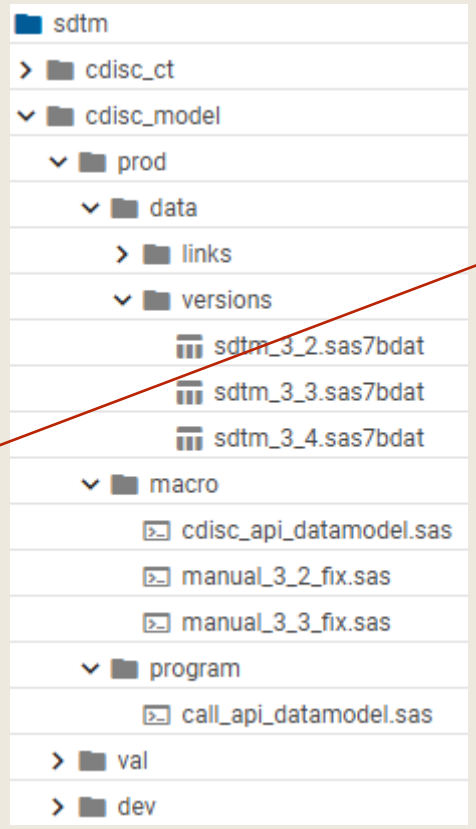


Standards area



Standards area

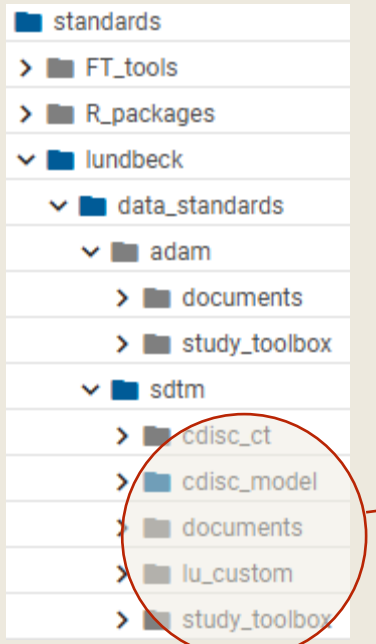


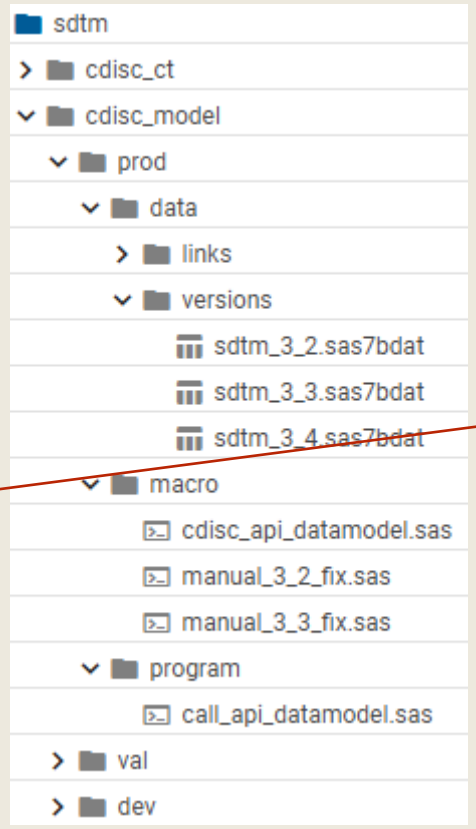


Normal / intuitive hierarchy:

1) CDISC
2) Lu_custom
3) Trial level

Standards area





Normal / intuitive hierarchy:

1) CDISC
2) Lu_custom
3) Trial level

Flex hierarchy:

1) Trial
2) Lu_std
3) CDISC

Standards area

```
/*************************************************************************************
PROGRAM NAME  :   cdisc_api_datamodel.sas
AUTHOR        :   MTAD
DESCRIPTION   :   Update the SDTM Library standards with newest SDTM data model.
                  This should be done every time a new version of the standard is released.
*************************************************************************************/
%macro cdisc_api_datamodel (version = );

  libname sdtm_std "standards/lundbeck/data_standards/sdtm/cdisc_model/&_exec_lifecycletype./data/versions";

  %** Make HTTP friendly versions;
  %let pri_v = %substr(&version.,1,1);
  %let sec_v = %substr(&version.,3,1);

  %** Get new SDTMIG version;
  filename resp temp;

  proc http
    url    = "https://library.cdisc.org/api/mdr/sdtmig/&pri_v.-&sec_v."
    method = "GET"
    out    = resp;
    headers "Cache-Control" = "no-cache"
            "api-key"       = " -- PUT ENTEPRISE API KEY HERE -- "
            "Accept"        = "application/json";
  run;

  libname cdisc JSON fileref=resp;

  %** Data recevied from API call is structured with some standard variables (one of a kind like the DM RFxxDTC)
      some class specific variable, which will be listed as --XXXXX.
      Need to create domain specific version of each class variable (replace -- with domain abbreviations);
```

Standards area

```
%** Grab the data model connected with user provided IG and download.
    This is used to create class variable for appliacble domains and also
    check usage restriction rules etc.;

proc sql noprint;
  select scan(href,-1,"/")
  into :model_v trimmed
  from cdisc._links_model
  ;
quit;

filename resp2 temp;

proc http
  url    = "https://library.cdisc.org/api/mdr/sdtm/&model_v."
  method = "GET"
  out    = resp2;
  headers "Cache-Control" = "no-cache"
          "api-key"       = " -- PUT ENTEPRISE API KEY HERE -- "
          "Accept"        = "application/json";
run;

libname cdisc2 JSON fileref=resp2;
```

Standards area

```
data class_vars01;
  %** New structure introduced in 3.4 where usage restriction can be used to make a more compliant output;
  set cdisc2.classes_classvariables
  %if &version. > 3.3 %then %do; (where=(usageRestrictions not contains ("Not in human clinical trials"))) %end;
  ;
run;

%** variables already described in IG causing metadata and sorting issues;
if name="--OBJ" and dataset in ("FA" "SR") then delete;
%** variables prohibited from use according to P21 (though not usage restriction text);
if dataset="DS" and variable in ("DSENDTC" "DSENDY" "DSENINT" "DSENRF" "DSENRTPT" "DSENTPT") then delete;
if dataset="FA" and variable in ("FACHSTDY" "FASTDTC" "FASTDY" "FAXSTDY") then delete;
if dataset="MB" and variable in ("NHOID") then delete;
if dataset="SR" and variable in ("SRCHSTDY" "SRSTDTC" "SRSTDY" "SRXSTDY") then delete;

%** Check usage restrictions to remove unwanted classvar creation, only available 3.4 and later;
%if &version. > 3.3 %then
  %do;
    %** variable only intended for a single domain not an entire class;
    if index(upcase(usageRestrictions), "DOMAIN ONLY") then
      do;
        if substr(usageRestrictions,1,2) ^= Dataset then delete;
      end;
    %*Known special cases;
    else if usageRestrictions = "CP, IS, and LB domains only"
      and Dataset not in ("CP" "IS" "LB") then delete;
    else if usageRestrictions = "Only in Findings class specimen-based domains: BS, CP, GF, IS, LB, MB, MS, MI, PC, PP"
      and Dataset not in ("BS" "CP" "GF" "IS" "LB" "MB" "MS" "MI" "PC" "PP") then delete;
    %** Not to be used in specific domain or class;
    else if index(upcase(usageRestrictions), "NOT IN") then
      do;
        %** Not in domain[s];
        if substr(usageRestrictions,8,2)=Dataset then delete;
        else
          do;
            %** Additional known special cases - keep in RS because it is listed IG vars;
            if usageRestrictions = "Not in QS, FT, and clinical classifications use case of RS"
              and Dataset in ("QS" "FT") then delete;
            else if usageRestrictions = "Not in Findings class domains"
              and class_name="Findings" then delete;
          end;
      end;
  %end;
```

Trial Level Utilities

1. One purpose per utility (small and specific)
2. indata= & outdata= macro parameters
3. Debug = Y/N (deactivate clean up and extend logs)

Trial Level Utilities

```
%createshells(domain=DM, spec=N);

data DM_01;
  if 0 then set DM_00;
  retain DOMAIN "&dom." STUDYID "&study_name.";
  set crf.dm (drop=studyid siteid);
  format _all_;
  informat _all_;
  ** Do not make any data manipulation in this step
     because of the double set statement;
run;

** Main DM build;
data DM_10 (keep = &domvar. DMRNDNUM RACEOTH RFPDESC) ;
  set DM_01;
  SITEID   = upcase(SITE);
  SUBJID   = tableLookup("crf.sc","dmscrnum","subject",subject);
  USUBJID  = catx('-', STUDYID,SITEID,SUBJID);
  DMRNDNUM = tableLookup("crf.dm_rand","dmrndnum","subject",subject);
  ** Reading RF--DTC data from the appropriate CRF datasets;
  RFICDTC  = tableLookup("crf.ds_in","ISOdatetime(incondat_raw,'')","subject",subject);
  RFPENDTC = tableLookup("work.rfpendtc","ISOdatetime(RFPENDTC,'')","subject",subject);
  RFPDESC  = tableLookup("work.rfpendtc","RFPDESC","subject",subject);
  ** Reading first, last doses from the temporary dataset above. These are also start, end reference dates;
  RFXSTDTC = tableLookup("work.exstdtc","RFXSTDTC","subject",subject);
  RFXENDTC = tableLookup("work.exendtc","RFXENDTC","subject",subject);
  ** Death info ;
  DTHDTC   = tableLookup("crf.sae(where=(aesdth_std='Y'))","ISOdatetime(dthdat_raw,'')","subject",subject);
  DTHFL    = tableLookup("crf.sae(where=(aesdth_std='Y'))","aesdth_std","subject",subject);
  ** Assign variables from crf.dm ;
  SEX      = upcase(sex_std);
  RACE     = upcase(race);
  if RACE  = "BLACK" then RACE="BLACK OR AFRICAN AMERICAN";
  ETHNIC   = upcase(ethnic);
  COUNTRY  = put(substr(sitenumber, 1, 2),$country.);
  *** ..... ;
run;

** Remove empty permissable variables;
%empty_var(indata=DM_10, outdata=DM_20);
** Create supp domain;
%gen_supp(indata=DM_30, outdata=DM_40);

** Finalise;
%mksdtm(indata=DM_40, outdata=sdtm.DM);
%mksdtm(indata=SUPPDM_40, domain=SUPPDM, outdata=sdtm.SUPPDM);
```

Trial Level Utilities

```
%createshells(domain=DM, spec=N);

data DM_01;
  if 0 then set DM_00;
  retain DOMAIN "&dom." STUDYID "&study_name.";
  set crf.dm (drop=studyid siteid);
  format _all_;
  informat _all_;
  ** Do not make any data manipulation in this step
     because of the double set statement;
run;

** Main DM build;
```

```
=== SDTM Summary ===============================================================================
Output dataset sdtm.DM created with 1334 observations.
No warnings issued.
================================================================================================
```

```
  RFICDTC  = tableLookup("crf.ds_in","ISOdatetime(incondat_raw,'')","subject",subject);
  RFPENDTC = tableLookup("work.rfpendtc","ISOdatetime(RFPENDTC,'')","subject",subject);
  RFPDESC  = tableLookup("work.rfpendtc","RFPDESC","subject",subject);
  ** Reading first, last doses from the temporary dataset above. These are also start, end reference dates;
  RFXSTDTC = tableLookup("work.exstdtc","RFXSTDTC","subject",subject);
  RFXENDTC = tableLookup("work.exendtc","RFXENDTC","subject",subject);
  ** Death info ;
  DTHDTC   = tableLookup("crf.sae(where=(aesdth_std='Y'))","ISOdatetime(dthdat_raw,'')","subject",subject);
  DTHFL    = tableLookup("crf.sae(where=(aesdth_std='Y'))","aesdth_std","subject",subject);
  ** Assign variables from crf.dm ;
  SEX      = upcase(sex_std);
  RACE     = upcase(race);
  if RACE  = "BLACK" then RACE="BLACK OR AFRICAN AMERICAN";
  ETHNIC   = upcase(ethnic);
  COUNTRY  = put(substr(sitenumber, 1, 2),$country.);
  *** ..... ;
run;

** Remove empty permissable variables;
%empty_var(indata=DM_10, outdata=DM_20);
** Create supp domain;
%gen_supp(indata=DM_30, outdata=DM_40);

** Finalise;
%mksdtm(indata=DM_40, outdata=sdtm.DM);
%mksdtm(indata=SUPPDM_40, domain=SUPPDM, outdata=sdtm.SUPPDM);
```

Trial Level Utilities

```
%createshells(domain=DM, spec=N);

data DM_01;
  if 0 then set DM_00;
  retain DOMAIN "&dom." STUDYID "&study_name.";
  set crf.dm (drop=studyid siteid);
  format _all_;
  informat _all_;
  ** Do not make any data manipulation in this step
     because of the double set statement;
run;

** Main DM build;
```

```
=== SDTM Summary ===============================================================================
Output dataset sdtm.DM created with 1334 observations.
No warnings issued.
================================================================================================
```

```
RFICDTC  = tableLookup("crf.ds_in","ISOdatetime(incondat_raw,'')","subject",subject);
RFPENDTC = tableLookup("work.rfpendtc","ISOdatetime(RFPENDTC,'')","subject",subject);
```

```
=== SDTM Summary ===============================================================================
Output dataset sdtm.QS created with 333759 observations.
Warning messages : 6
================================================================================================
WARNING : EPOCH <SCREENING ATTEMPT 1> not found in extensible codelist C99079 'Epoch'
WARNING : QSCAT <MBS > not found in extensible codelist C100129 'Category of Question'
WARNING : QSCAT <MSQ > not found in extensible codelist C100129 'Category of Question'
WARNING : QSCAT <MiCOAS-COG > not found in extensible codelist C100129 'Category of Question'
WARNING : QSCAT <mMIDAS > not found in extensible codelist C100129 'Category of Question'
WARNING : QSMETHOD <LIKERT SCALE 4-POINT> not found in extensible codelist C158113 'Method of Test or Examination'
================================================================================================
```

```
** Create supp domain;
%gen_supp(indata=DM_30, outdata=DM_40);

** Finalise;
%mksdtm(indata=DM_40, outdata=sdtm.DM);
%mksdtm(indata=SUPPDM_40, domain=SUPPDM, outdata=sdtm.SUPPDM);
```

Trial Level Utilities

```
** creates VISIT(NUM+DY) by matching current domain
   with SV. Important to have a solid SV in place and dates/visits from raw;
%gen_visit(indata=VS_41, outdata=VS_50, dtcvar=VSDTC, rawvisit=FolderName, visfmt=visit);

** Sets epoch where possible. Manually inspect special cases
   where EPOCH is missing or not set as expected
   (e.g. when protocol knowledge is needed and timevar not available);
%gen_epoch(indata=VS_50, outdata=VS_60, dtcvar=VSDTC);

** Create standardized variables --STRES(C+N+U)
   Ensure special cases are updated in study_details.xlsx;
%gen_stres(indata=VS_60, outdata=VS_70);

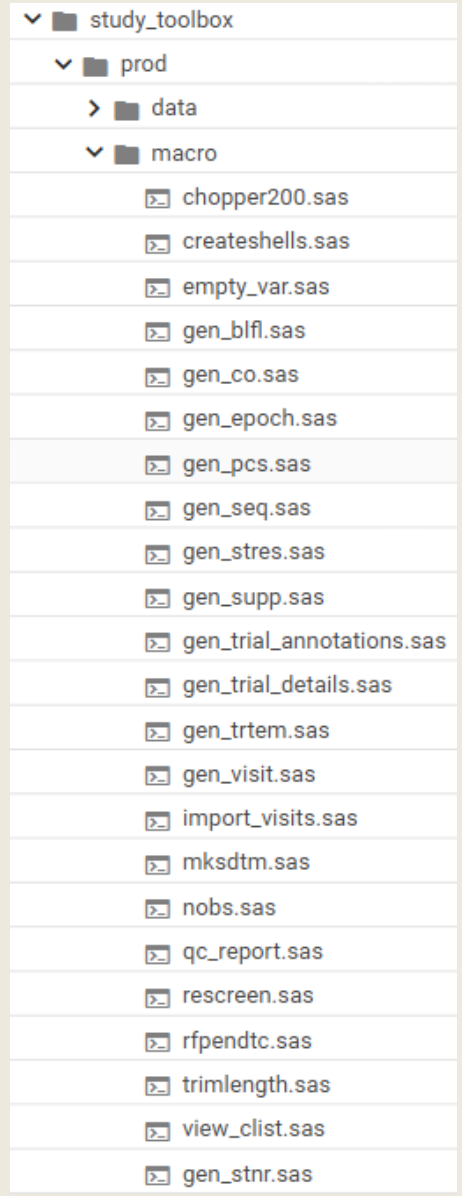
** Set baseline flag, default is --BLFL.
   e.g. outvar=LOBXFL could be set, if IG 3_4 is used;
%gen_blfl(indata=VS_70, outdata=VS_80, outvar=%str(&dom.LOBXFL));

** Create SEQuence variable and test key for uniqueness
   set create shells to run on spec to automatically test
   key entered in spec for uniqueness (submission requirement);
%gen_seq(indata=VS_80, outdata=VS_90, key=&domkey.);

** Remove all blank CDISC permissable variables;
%empty_var(indata=VS_90, outdata=VS_100);

** prepare supplemental qualifiers if applicable;
%gen_supp(indata=VS_100, outdata=VS_110);

*****************************************************************
                        Finalise
*****************************************************************;
%mksdtm(indata=VS_110, outdata=sdtm.VS);
%mksdtm(indata=SUPPVS_110, domain=SUPPVS, outdata=sdtm.SUPPVS);
```



Submission

HO (Healthcare Encounters) FA (Findings About Events or Interventions)
AE (Adverse Events) DM (Demographics)

20297A_ver 1.00_21MAR24_PROD: PDF

Form: Serious Adverse Event

Generated On: 09 Apr 2024 09:13:30

SAE number:

Is the SAE start date the same as the **PRIMARY** AE start date? Yes ☐ No ☐

If no, fill in actual SAE start date: FADTC when FATESTCD = SERIOUS

Seriousness criteria

Death: AE.AESDTH Yes ☐ No ☐

Life threatening: AE.AESLIFE Yes ☐ No ☐

Required or prolonged hospitalization: AE.AESHOSP Yes ☐ No ☐

Persistent or significant disability/incapacity: AE.AESDISAB Yes ☐ No ☐

A congenital anomaly/birth defect: AE.AESCONG Yes ☐ No ☐

Other important medical events: AE.AESMIE Yes ☐ No ☐

Hospitalization HO.HOTERM

Date of hospitalization: HO.HOSTDTC

Date of discharge: HO.HOENDTC

Hospitalization ongoing: HO.HOENRTPT

Death

Date of death: DM.DTHDTC

Standard	Type	Status	Documentation
ADaMIG 1.1	IG	Final	This trial uses ADaM IG version 1.3
CDISC/NCI SDTM 2025-09-26	CT	Final	This trial uses latest SDTM terminology at trial start, which is 2024-09-27.
CDISC/NCI ADaM 2024-09-27	CT	Final	This trial uses latest ADaM terminology at trial start, which is 2026-03-27.

Datasets

Dataset	Description	Class	Structure	Purpose	Keys	Documentation	Location
ADAE [ADaMIG 1.1]	Adverse Event Analysis Dataset	OCCURRENCE DATA STRUCTURE	One record per participant per adverse event per severity/seriousness	Analysis	USUBJID, AESPID, ASTDT, FADTC		adae.xpt
ADCAGII [ADaMIG 1.1]	Caregiver CGII Analysis Dataset	BASIC DATA STRUCTURE	One record per participant per parameter per assessment	Analysis	USUBJID, APHASEN, AVISITN, ADTM		adcagii.xpt
ADCE [ADaMIG 1.1]	Clinical Events Analysis Dataset	ADAM OTHER	One record per participant per day per seizure	Analysis	USUBJID, ADT, CESPID, SEIZID, CLUSTID, TERM, DURATION		adce.xpt
ADCGII [ADaMIG 1.1]	CGI - Improvement Analysis Dataset	BASIC DATA STRUCTURE	One record per participant per parameter per assessment	Analysis	USUBJID, APHASEN, AVISITN, ADTM, PARAMCD		adcgii.xpt
ADCGIICE [ADaMIG 1.1]	CGI - Improvement Analysis Dataset ICE	BASIC DATA STRUCTURE	One record per participant per parameter per assessment	Analysis	USUBJID, APHASEN, AVISITN, ADTM, PARAMCD		adcgiice.xpt
ADCGIS [ADaMIG 1.1]	CGI - Severity Analysis Dataset	BASIC DATA STRUCTURE	One record per participant per parameter per assessment	Analysis	USUBJID, APHASEN, AVISITN, ADTM, PARAMCD		adcgis.xpt
ADCM [ADaMIG 1.1]	Concomitant Medication	OCCURRENCE DATA	One record per participant per concomitant	Analysis	USUBJID, CMSEQ		adcm.xpt

Submission – acrf.pdf

```
data CDISC_anno_90;
  set CDISC_anno_80;
  length bg_col txt_col txt_weight $30 fontsize 8;
  %** color code multiple domains per page;
  if dom_index = 0      then bg_col = "#BFBFBF"; *GREY (Not submitted etc.);
  else if dom_index = 1 then bg_col = "#BFFFFF"; *AQUA LIGHT;
  else if dom_index = 2 then bg_col = "#FFFFAA"; *YELLOW LIGHT;
  else if dom_index = 3 then bg_col = "#BFFFBF"; *GREEN LIGHT;
  else if dom_index = 4 then bg_col = "#FFBFAA"; *PEACH LIGHT;
  else if dom_index = 5 then bg_col = "#F7C5A9"; *PURPLE LIGHT;
  else if dom_index > 5 then putlog "W" "ARNING: more than 5 domains on one page. Add additional HTML color codes";
  %** std visual for domain annotations;
  if dlabel = label and dlabel^="" then
    do;
      txt_col = "#000000";
      txt_weight = "bold";
      fontsize = 11;
      border = "S";
    end;
    else
      do;
        txt_col = "#000000";
        txt_weight = "normal";
        fontsize = 9;
      end;
run;
```

```
%** XML codeline box attributes (color, size, position etc.);
      data CDISC_anno_100;
        length line $ 500;
        set CDISC_anno_90 end=last;
        if _n_ = 1 then
          do;
            line = "<?xml version=""1.0"" encoding=""UTF-8""?>";
            output;
            line = "<xfdf xmlns=""http://ns.adobe.com/xfdf/"" xml:space=""preserve"">";
            output;
            line = "  <annots>";
            output;
          end;
        line = "    <freetext rect=""" || strip(coords) || """ rotation=""" || strip(rotation) ||""""";
        output;
        line = "      opacity=""1"" color=""" || strip(bg_col) || """ flags=""print""";
        output;
        line = "      subject=""" || strip(label) || """ page=""" || strip(page) || """>";
        output;
        line = "      <contents-richtext>";
        output;
        line = "        <body xmlns=""http://www.w3.org/1999/xhtml""";
        output;
        line = "          xmlns:xfa=""http://www.xfa.org/schema/xfa-data/1.0/""          ";
        output;
        line = "          xfa:APIVersion=""Acrobat:7.0.0""";
        output;
        line = "          xfa:spec=""2.0.2"" style=""font-size:"||strip(fontsize)||".0pt;font-family:Arial,sans-
serif;""><p><span style=""font-family:Arial; font-weight:" || strip(txt_weight) || ";"">" || strip(label) ||
"</span></p></body>";
        output;
        line = "      </contents-richtext>";
        output;
        line = "      <defaultstyle>font: Arial,sans-serif "||strip(fontsize)||".0pt; text-align:left; color:"||
strip(txt_col)||" </defaultstyle>";
        output;
        line = "      <border width=""1"" style="""|| border ||"""/>";
        output;
        line = "    </freetext>";
        output;

        if last then
          do;
            line = "  </annots>";
            output;
            line = "</xfdf>";
            output;
          end;
      run;

      data _null_ ;
        set CDISC_anno_100 (keep=line where=(line ne ''));
        file "&anno_xfdf.";
        put line;
      run;
```

Submission – define.xml

```
proc import datafile="&spec_path." out=datasets (where=(dataset^=""))
dbms=xlsx replace;
sheet='Datasets';
run;
proc import datafile="&spec_path." out=common_ dbms=xlsx replace;
sheet='CommonVariables';
run;
proc import datafile="&spec_path." out=variables_1 dbms=xlsx replace;
sheet='Variables';
run;
proc import datafile="&spec_path." out=valuelvl_1 dbms=xlsx replace;
sheet='ValueLevel';
run;
proc import datafile="&spec_path." out=codelists_1(where=(Codelist_Name^=""
and Define="Y")) dbms=xlsx replace;
sheet='Codelists';
run;
proc import datafile="&spec_path." out=standards dbms=xlsx replace;
sheet='Standards';
run;
proc import datafile="&spec_path." out=documents dbms=xlsx replace;
sheet='Documents';
run;
```

Submission – define.xml

```
**-------------------------------------------------------------**
**  Get actual SDTM/ADaM data information about length/digits  **
**    - if user specified data_read = Y                        **
**-------------------------------------------------------------**;
%if %lowcase(&data_read) = y %then %do;
```

```
** variable tab;
 if var then do;
   if lowcase(origin)="derived" then do; com_mt_pre_oid=cats("MT.",dataset,".",variable); source="Sponsor"; end;
   else if lowcase(origin)="assigned" then do; com_mt_pre_oid=cats("COM.",dataset,".",variable); source="Sponsor"; end;
   else if lowcase(origin)="predecessor" then do; com_mt_pre_oid=source_derivation; source=""; end;
 else if origin="" then do; com_mt_pre_oid = cats("VL.",dataset,".",variable); end;
 end;
 ** valuelevel tab (oid so no MT. COM. needed);
 else if val then do;
   if lowcase(origin) in ("derived" "assigned") then do; com_mt_pre_oid=strip(cats(dataset,'.',variable,'.',tranwrd(strip(where_clause),' ','.')));
source="Sponsor"; end;
   else if lowcase(origin)="predecessor" then do; com_mt_pre_oid=strip(cats(dataset,'.',variable,'.',tranwrd(strip(where_clause),' ','.')));
source=""; end;
 end;
 ** multiple values required to be entered in parenthesis with comma seperator, so remove from OID creation;
 com_mt_pre_oid = compress(com_mt_pre_oid,"(),");
 ** markup;
 source_derivation=tranwrd(source_derivation,"&",%nrstr("&amp;"));
 source_derivation=tranwrd(source_derivation,"<",%nrstr("&lt;"));
 source_derivation=tranwrd(source_derivation,">",%nrstr("&gt;"));
 source_derivation=tranwrd(source_derivation,"'",%nrstr("&apos;"));
 ** LF+CR;
 source_derivation = tranwrd(source_derivation, '0D0A'x, ' ');
 source_derivation = tranwrd(source_derivation, '0A'x, ' ');
 source_derivation = tranwrd(source_derivation, '0D'x, ' ');
 ** smart quotes;
 source_derivation = translate(source_derivation, '"', '"″');
```

Submission – define.xml

```
** define.xml - Initiation (1) ;
data define_intro_1;
length line $ 2999;
line=""<?xml version=""1.0"" encoding=""UTF-8""?>";output;
line="<?xml-stylesheet type=""text/xsl""
href=""define&def_vs..xsl""?>";output;
line="<ODM";output;
line="     xmlns:xlink=""http://www.w3.org/1999/xlink""";output;
line="     xmlns=""http://www.cdisc.org/ns/odm/v1.3""";output;
line="
xmlns:def=""http://www.cdisc.org/ns/def/v&def_pri_vs..&def_sec_vs""";output;
line="     ODMVersion=""1.3.2""";output;
line="     FileType=""Snapshot""";output;
line="     FileOID=""&StudyName.""";output;
line="
CreationDateTime="""||"%sysfunc(date(),yymmdd10.)T%sysfunc(time(),tod8.)"||'"
';output;
line="     SourceSystem=""SAS""";output;
line="     SourceSystemVersion=""9.9.9""";output;
line="     def:Context=""Submission"">";output;
line="   <Study OID=""&StudyName.""">";output;
line="      <GlobalVariables>";output;
line="         <StudyName>&StudyName.</StudyName>";output;
line="
<StudyDescription>&StudyDescription.</StudyDescription>";output;
line="         <ProtocolName>&ProtocolName.</ProtocolName>";output;
line="      </GlobalVariables>";output;
line="      <MetaDataVersion OID=""MDV.&StudyName."" Name=""Study &StudyName.
Data Definitions""
def:DefineVersion=""&def_pri_vs..&def_sec_vs..10"">";output;
run;
```

Submission – define.xml

```
** define.xml - standards part (2);
data define_standards_2_1;
  set standards (where=(upcase(type)="IG")) end=eof;
  by id;
  length line $ 2999;
  if _n_=1 then do;
    line="         <def:Standards>"; output;
  end;
  line="           <def:Standard OID=""STD."||strip(id)||""" Name="""||strip(name)||"""
Type="""||strip(type)||"""; output;
  line="             Version="""||strip(version)||""" Status=""Final""
def:CommentOID=""COM."||strip(id)||"""/>"; output;
  keep line;
run;

data define_standards_2_2;
  set standards (where=(upcase(type)="CT")) end=eof;
  by id;
  length line $ 2999;
  line="           <def:Standard OID=""STD."||strip(id)||""" Name="""||strip(name)||"""
Type="""||strip(type)||"""; output;
  line="             PublishingSet="""||strip(publish)||""" Version="""||strip(version)||""" Status=""Final""
def:CommentOID=""COM."||strip(id)||"""/>"; output;
  if eof then do;
    line="         </def:Standards>"; output;
  end;
  keep line;
run;
```

Submission – define.xml

```
** define.xml - suppl.documents (3);
* ....;
** define.xml - ValueLevel + WhereClause (4);
* ....;
** define.xml - dataset and variable part (5);
* ....;
** define.xml - item definitions (from variables tab + valuelevel)(6);
* ....;
** define.xml - codelists (7);
* ....;
** define.xml - methods from variables and valuelevel tabs (8);
* ....;
** define.xml - comments from variables and valuelevel tabs (9);
* ....;

data define_closure_10;
  set documents end=last;
  line="         <def:leaf ID=""LF."||strip(id)||"""
xlink:href="""||strip(href)||""">";output;
  line="            <def:title>"||strip(title)||"</def:title>";output;
  line='         </def:leaf>';output;
  if last then do;
    line="      </MetaDataVersion>";output;
    line="   </Study>";output;
    line="</ODM>";output;
  end;
  keep line;
run;
```

Submission – define.xml

```
** define.xml creation;
data _null_;
  set define_intro_1
      define_standards_2_1
      define_standards_2_2
      define_suppdoc_3
      define_valuelvl_4_1
      define_where_4_2
      define_data_5
      define_itemdef_6_1
      define_itemdef_6_2
      define_codelist_7_1
      define_codelist_7_2
      define_codelist_7_3
      define_methods_8_1
      define_methods_8_2
      define_comments_9_1
      define_comments_9_2
      define_closure_10
  ;
  file "&path./define.xml" encoding='utf-8' nobom;
  put line :$char2999.;
  ;
run;
```

Lundbeck

