

Rule-Based Auto-SDTM Mapping Software in Python

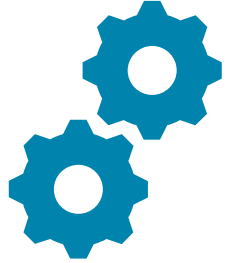
Automatically Generates Executable SAS or R Programs

March 2026

Agenda Overview

- Background & Motivation
- System Architecture and Technology Stack
- Related Metadata and Documents
- Automated SDTM Mapping Workflow
- Customization and Flexibility for Sponsors
- Case Studies: Generated SAS Code for SDTM Domains
- Future Directions

Background & Solution



Background

Repetitive Patterns and Opportunities for automation in SDTM mapping

Complexity of Evolving Standards and Increasing Data Variety

Industry Approaches to SDTM auto-mapping - SAS Macros and AI Assistants



ClinChoice Automation Solution

Metadata-driven & Rule-based with Python Implementation

AI Assistant

Supporting Output of SAS, R and more

General Framework & Approaches

Standardized, Accurate, and Traceable SDTM Mapping Processes

Close to Current Practice - less Learning Curve, less Operation Change

System Design

- **Core Technology Stack**

The software uses Python with pandas, Lex-Yacc (sly) parser, Excel tools, and AI language libraries to build its core functionality.

- **Command Line Integration**

Python scripts run from the command line enabling easy integration with web servers, AI agents, and standalone apps.

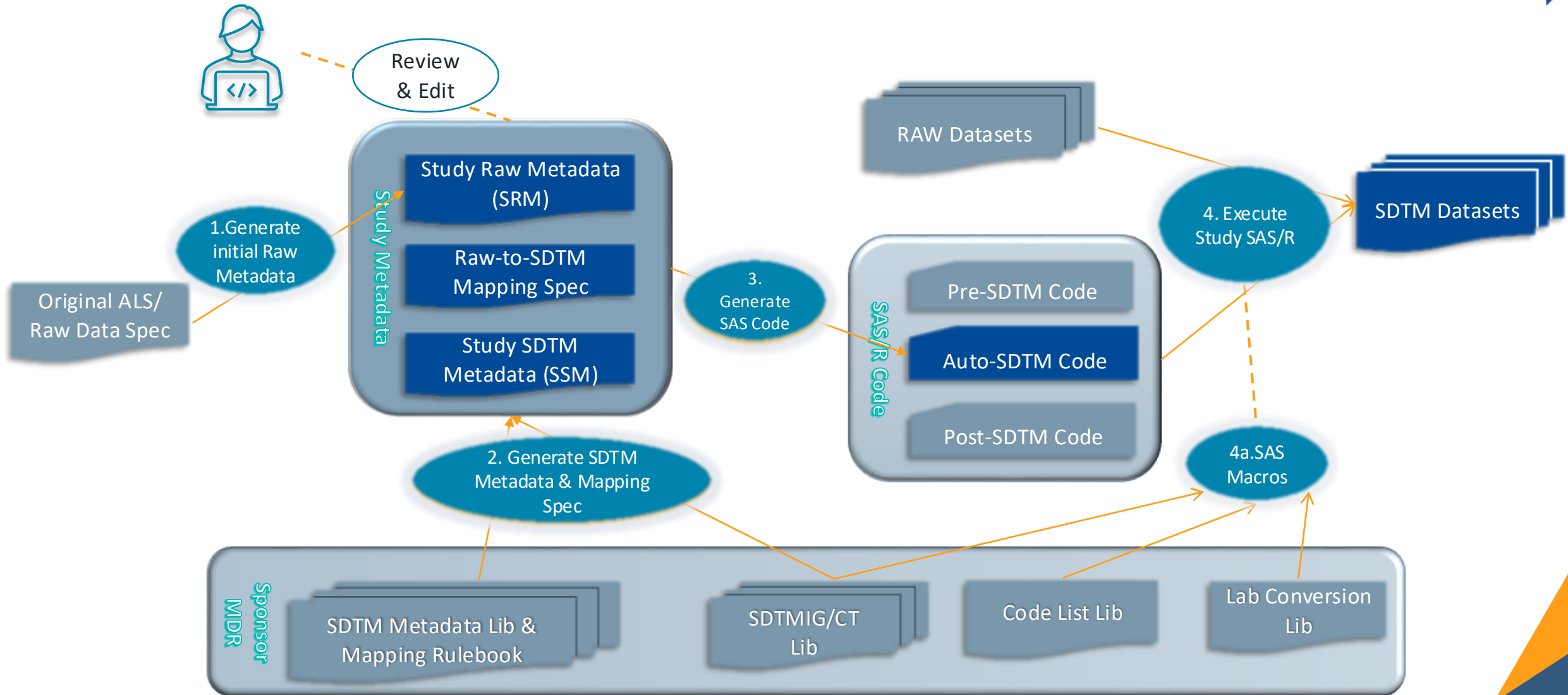
- **Document and Format Standards**

All documents are generated in Excel format adhering to industry standards for data interchange and reporting.

- **SAS and R Integration**

The software supports SAS and R integration by executing scripts within the process flow for advanced analytics.

High-level SDTM Mapping Process Flow



Related Metadata and Documents

- CDSIC SDTM IGs & Controlled Terminology
- Sponsor SDTM Rulebook

RAW				SDTM			
Dataset	Variable	Label	Map Rule	Dataset	Variable	Label	Core
AE	STUDY	Study Code	COPY()	AE	STUDYID	Study Identifier	Req
AE			"AE"	AE	DOMAIN	Domain Abbreviation	Req
AE	STUDY SUBJECT	Study Code Enrolment Code	USUBJID()	AE	USUBJID	Unique Subject Identifier	Req
AE	AENO	AE Number	CONCAT('AE-', AENO)	AE	AESPID	Sponsor-Defined Identifier	Perm
AE	AETERM	Adverse Event Reported Term	COPY()	AE	AETERM	Reported Term for the Adverse Event	Req
AE	LLT_NAME	MedDRA Lowest Level Term Name	COPY()	AE	AELLT	Lowest Level Term	Exp
AE	LLT_CODE	MedDRA Lowest Level Term Code	COPY()	AE	AELLTCD	Lowest Level Term Code	Exp
AE	PT_NAME	MedDRA Preferred Term Name	COPY()	AE	AEDECOD	Dictionary-Derived Term	Req
AE	PT_CODE	MedDRA Preferred Term Code	COPY()	AE	AEPTCD	Preferred Term Code	Exp
AE	HLT_NAME	MedDRA High Level Term Name	COPY()	AE	AEHLT	High Level Term	Exp
AE	HLT_CODE	MedDRA High Level Term Code	COPY()	AE	AEHLTCD	High Level Term Code	Exp
AE	HLGTNAME	MedDRA High Level Group Term Name	COPY()	AE	AEHLGT	High Level Group Term	Exp
AE	HLGTCODE	MedDRA High Level Group Term Code	COPY()	AE	AEHLGTC	High Level Group Term Code	Exp
AE	SOC_NAME	MedDRA System Organ Class Name	COPY()	AE	AEBODSYS	Body System or Organ Class	Exp
AE	SOC_CODE	MedDRA System Organ Class Code	COPY()	AE	AEBDSYCD	Body System or Organ Class Code	Exp
AE	SOC_NAME	MedDRA System Organ Class Name	COPY()	AE	AESOC	Primary System Organ Class	Exp
AE	SOC_CODE	MedDRA System Organ Class Code	COPY()	AE	AESOC	Primary System Organ Class Code	Exp
AE	AESEV AEC01SEV AEC02SEV	AE Intensity AE Intensity AE Intensity	MAX(UPDECODE(AESEV), UPDECODE(AEC01SEV), UPDECODE(AEC02SEV))	AE	AESEV	Severity/Intensity	Perm
AE	AESEVMAX	Maximum AE Intensity	UPDECODE()	AE	AESEV	Severity/Intensity	Perm
AE	AESER	Serious Adverse Event	UPDECODE()	AE	AESER	Serious Event	Exp
AE	AEACN AEC01AC	Action Taken, Investigational Product Action Taken, Investigational Product	if AEC01AC ne "" then 'MULTIPLE' else if AEACN = '4' then 'DRUG WITHDRAWN' else UPDECODE(AEACN)	AE	AEACN	Action Taken with Study Treatment	Exp

Related Metadata and Documents

- Raw Metadata Specification File
- Unified Study Raw Metadata (SRM)

Dataset ▾	Module ▾	Variable ▾	Variable Label ▾	Type ▾	Collection Length ▾	Format ▾	Current Version ▾	Source ▾
AE	AE	STUDY	Study Code	C	11		AZ2403 2024-03-25	CRF
AE	AE	SUBJECT	Enrolment Code	C	8		AZ2403 2024-03-25	CRF
AE	AE	PART	Study Part Code	C	1		AZ2403 2024-03-25	CRF
AE	AE	LINE	Line Number	N	3		AZ2403 2024-03-25	CRF
AE	AE	AEYN	Any Adverse Event	C	6	\$NY	AZ2403 2024-03-25	CRF
AE	AE	AENO	AE Number	N	3		AZ2403 2024-03-25	CRF
AE	AE	AETERM	Adverse Event Reported Term	C	200		AZ2403 2024-03-25	CRF
AE	AE	AESTDAT	Adverse Event Start Date	C	10		AZ2403 2024-03-25	CRF
AE	AE	AEENDAT	Adverse Event End Date	C	10		AZ2403 2024-03-25	CRF
AE	AE	AEOUT	Outcome of Adverse Event	C	6	\$OUT	AZ2403 2024-03-25	CRF
AE	AE	AESER	Serious Adverse Event	C	6	\$NY	AZ2403 2024-03-25	CRF
AE	AE	IP	Investigational Product	C	20		AZ2403 2024-03-25	CRF
AE	AE	AEACN	Action Taken, Investigational Product	C	2	\$AEACN	AZ2403 2024-03-25	CRF
AE	AE	AESEVMAX	Maximum AE Intensity	C	6	\$AESEV	AZ2403 2024-03-25	CRF
AE	AE	AEREL	Reasonable Possibility AE Caused by IP	C	6	\$NY	AZ2403 2024-03-25	CRF
AE	AE	AEDIS	AE Caused Subject to Withdraw from Study	C	6	\$NY	AZ2403 2024-03-25	CRF
AE	AE	AEPROSSP	Provoked by Specific Study Procedure	C	6	\$NY	AZ2403 2024-03-25	CRF
AE	AE	AECONTRT	Concomitant or Additional Trtmnt Given	C	6	\$NY	AZ2403 2024-03-25	CRF
AE	AE	LT_CODE	MedDRA Lowest Level Term Code	N	6		AZ2403 2024-03-25	CRF

Related Metadata and Documents

- Unified Study SDTM Metadata (SSM)

Dataset	Variable	Label	Type	Length	Core	Role	Format	Key	Seq	Origin
AE	STUDYID	Study Identifier	C	21	req	Identifier		1	1	Protocol
AE	DOMAIN	Domain Abbreviation	C	8	req	Identifier	(DOMAIN)		2	Derived
AE	USUBJID	Unique Subject Identifier	C	30	req	Identifier		2	3	Derived
AE	AESEQ	Sequence Number	N	8	req	Identifier			4	Derived
AE	AEGRPID	Group ID	C	40	perm	Identifier			5	Derived
AE	AEREFID	Reference ID	C	40	perm	Identifier			6	Derived
AE	AESPID	Sponsor-Defined Identifier	C	40	perm	Identifier			7	CRF
AE	AELNKID	Link ID	C	40	perm	Identifier			8	CRF
AE	AELNKGRP	Link Group ID	C	40	perm	Identifier			9	CRF
AE	AETERM	Reported Term for the Adverse Event	C	200	req	Topic			10	CRF
AE	AEMODIFY	Modified Reported Term	C	200	perm	Synonym Qualifier			11	Assigned
AE	AELLT	Lowest Level Term	C	200	exp	Variable Qualifier	MedDRA		12	Assigned
AE	AELLTCD	Lowest Level Term Code	N	8	exp	Variable Qualifier	MedDRA		13	Assigned
AE	AEDECOD	Dictionary-Derived Term	C	200	req	Synonym Qualifier	MedDRA	4	14	Assigned

Related Metadata and Documents

- Unified Study Raw-to-SDTM Mapping Specification (SRSMS)

A	B	C	D	E	F	G	H	I
Source Dataset	Source Variable	Source Label	Map Rule	Rule Validation	Map Definition	SDTM Dataset	SDTM Variable	
AE	AECONTRT	Concomitant or Additional Trtmt Given	DECODE()	Passed	The decoded value of AECONTRT	AE	AECONTRT	
AE	AESTDAT	Adverse Event Start Date	DTC(AESTDAT, AESTTIM)	Undefined: AE.AESTTIM	The date/time value of AESTDAT and AESTTIM in ISO8601 format	AE	AESTDTC	
AE	AESTTIM	Start Time of Adverse Event		Skipped		AE	AESTDTC	
AE	AEENDAT	Adverse Event End Date	DTC(AEENDAT, AEENTIM)	Undefined: AE.AEENTIM	The date/time value of AEENDAT and AEENTIM in ISO8601 format	AE	AEENDTC	
AE	AEENTIM	End Time of Adverse Event		Skipped		AE	AEENDTC	
AE	AEENDAT	Adverse Event End Date	if AEENDAT = "" and AEOUT in('C49498', 'C49495') then 'BEFORE' else if AEOUT = 'C48275' then 'COINCIDENT' else if AEOUT in('C49496', 'C49494') then 'ONGOING' else ""	Passed	If AEENDAT equals "" and AEOUT includes values ('C49498', 'C49495') then 'BEFORE' else if AEOUT equals 'C48275' then 'COINCIDENT' else if AEOUT includes values ('C49496', 'C49494') then 'ONGOING' else missing value	AE	AEENRTPT	
AE	AEOUT	Outcome of Adverse Event		Skipped		AE	AEENRTPT	
AE	AEENDAT	Adverse Event End Date	if AEENDAT ne "" then "" else if AEOUT in('C49498', 'C49496', 'C49495', 'C49494') then 'STUDY TERMINATION' else if AEOUT = 'C48275' then 'DEATH' else ""	Passed	If AEENDAT not equals "" then missing value else if AEOUT includes values ('C49498', 'C49496', 'C49495', 'C49494') then 'STUDY TERMINATION' else if AEOUT equals 'C48275' then 'DEATH' else missing value NOTE: The time point can be adjusted depending on study needs	AE	AEENTPT	
AE	AEOUT	Outcome of Adverse Event		Skipped		AE	AEENTPT	
AE	AEPROSSP	Provoked by Specific Study Procedure	DECODE()	Passed	The decoded value of AEPROSSP	SUPPAE	AEPROSSP	
AE	AEDIS	AE Caused Subject to Withdraw from	DECODE()	Passed	The decoded value of AEDIS	SUPPAE	AEWD	

Related Metadata and Documents

- **Expression and Logical Operators**

- Common expression and logical operators enable conditional checks and decision-making in data workflows.

- **Functions**

- Covering popular functions like STRIP, UPCASE, DECODE, UPDECODE, support efficient text processing and cleanup of clinical data.

- **Advanced Functions and Macros**

- Common SAS macros automate complex tasks like sequence generation and date/time conversion.

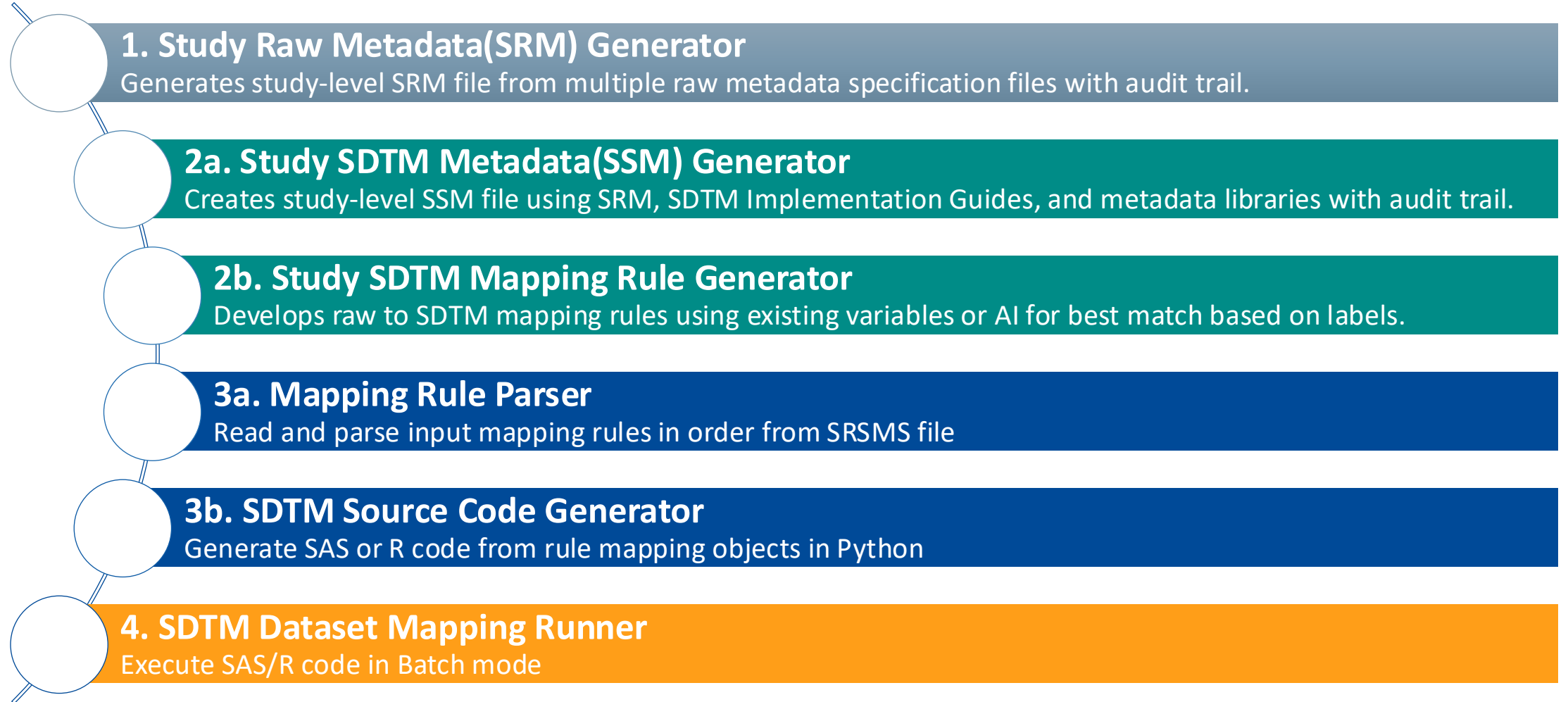
Mapping Expression	Alias	Description	Sample Expression	SAS Code
Expression Operators				
=, !=, >, <, >=, <=		Boolean operators for condition	Var2 = 'N' RAW1.VAR1>RAW2.VAR3	
+, -, *, /		Regular operators	RAW1.VAR1 * 2 *100 /3.2	
&, , !, and, or, not		Logical operators	VAR1>VAR3 & VAR1<=VAR4 VAR1>VAR3 and VAR1<=VAR4	VAR1>VAR3 and VAR1<=VAR4
IN, NOT IN		in statement	var1 in ('1','A') var1 Not in ('2','B')	var1 in ('1','A') var1 Not in ('2','B')
Functions				
ISNULL	missing	Is missing	ISNULL(RAW.VAR1) Missing(RAW.VAR1)	Missing(RAW.VAR1)
IFC		If-else function	IF(ISNULL(RAW.VAR1), raw1.var1, raw2.var2) IF(RAW.VAR1 != "", raw1.var1)	
COPY() COPY(<RAW variable>)		Copy source value to target based on variable type	copy() copy(RAW.VAR4)	Char to Char: strip(<RAW Variable>) Num to Num:<RAW Variable> Char to Num: input(<RAW Variable>, best.) Num to Char: strip(put(<RAW Variable>, best.))
USUBJID()		Concatenate STUDY, '/' and SUBJECT to USUBJID	USUBJID()	cats(STUDY, '/', SUBJECT)
DECODE() DECODE(<RAW Variable>) DECODE(<Format>) DECODE(<RAW Variable>, <Format>)	PUT, INPUT	Decode the RAW variable, or using RAW variable format, or using a specified format	DECODE() DECODE(<RAW Variable>) DECODE(<RAW Variable>, <Format>)	strip(putc(<RAW Variable>, raw_format(<RAW Variable>))) Decode the RAW variable using RAW variable format strip(put(<RAW Variable>, <Format>))

Related Metadata and Documents

- **Predefined Mapping Rules**

UPDECODE() UPDECODE(<RAW Variable>) UPDECODE(<Format>) UPDECODE(<RAW Variable>, <Format>) STRIP(), STRIP(Var1)		Decode (with uppercase) the RAW variable, or using RAW variable format, or using a specified format	DECODE() DECODE(raw_var1) DECODE(\$NY) DECODE(raw_var2, \$Y)	strip(putc(<RAW Variable>, raw_format(<RAW Variable>))) Decode the RAW variable using RAW variable format strip(put(<RAW Variable>, <Format>))
VARLABEL(<RAW Variable>)		Set to the RAW variable label		strip(vlabel(<RAW Variable>))
UPCASE(<RAW Variable>)	TOUPPER	Set to the RAW variable value in upper case	UPCASE(RAW.CMTRT)	upcase(strip(<RAW Variable>))
SCAN		separate parts of a character value. (split)	SCAN(RAW.VAR1, 2, "T")	SCAN(RAW.VAR1, 2, "T")
FIND		To locate a substring within a string.	FIND(RAW.VAR1, "XXXX")	FIND(RAW.VAR1, "XXXX")
DELETE_IF(condition)		skip record if the condition is met	delete_if(AETERM='')	if AETERM="" then delete;
CATX(<Variable1>, '<String1>', <Variable2>, ...)	Paste	Concatenates multiple character variables and/or strings	CATX(separator, RAW.OTHER, RAW.OTHERSP)	strip(<Variable1>) <String1> strip(<Variable2>) ...
CATS('<String1>', <Variable2>, ...)	Paste0, concat	Concatenates multiple character variables and/or strings	CATS(RAW.OTHER, RAW.OTHERSP)	strip(<Variable1>) <String1> strip(<Variable2>) ...
MIN(<Variable1/Value1>, <Variable2/Value2>, ...)		Get the lowest value of the listed values		min(<Variable1/Value1>, <Variable2/Value2>, ...)
Macros (%MACRO_NAME(...))				
%SEQ(<Variable1>, <Variable2>, ...)		Creates the Sequence Number		%seq(dataset_layout=<Variable1> <Variable2> ..., outvar_seq=WORK.<SDTM Dataset>.<SDTM Variable>);
DTC() DTC(<Date Variable>)	ISO_DATE	Creates the DTC from date variable		%dtc(INVAR_DAT=<Date Variable>, OUTVAR_DTC=<SDTM Variable>)
DTC(<Date Variable>, <Time Variable>)	ISO_DATETIME	Creates the DTC from date and time variables		%dtc(INVAR_DAT=<Date Variable>, INVAR_TIM=<Time Variable>, OUTVAR_DTC=<SDTM Variable>)

Modules for Automated SDTM Mapping Flow



Customization and Flexibility for Sponsors

- **Pre- and Post-process Code**

SDTM domains support custom pre- and post-processing using dataset-level SAS macros for special handling.

- **Customizable Mapping Rules**

Mapping rule definitions can be tailored by updating the mapping rule parser to fit sponsor needs.

- **Adjustable Code Generation**

Generated SAS/R code is modifiable using the updated SDTM Source Code Generator.

- **Modifiable Variable Formats**

Variable formats are defined in an easily editable SDTM metadata spreadsheet.

Case Studies: Generated SAS Code for AE

```
/****** START OF AUTO PROGRAM *****/

/* DATASET: AE_AE */
data AE_AE_TMP;
  length AEACN_SDTM AECONTRT_SDTM AEENDTC AEENRTPT AEENTPT AEOUT_SDTM AEREL_SDTM AESER_SDTM AESEV AESPID AESTDTC $200;
  set RAW.AE;
  STUDYID = strip(STUDY);
  DOMAIN = "AE";
  USUBJID = CATS(STUDY,"/",SUBJECT);
  AESPID = CATS('AE-',AENO);
  AETERM_SDTM = strip(AETERM);
  AELLT = strip(LLT_NAME);
  AELLTCD = LLT_CODE;
  AEDECOD = strip(PT_NAME);
  AEPTCD = PT_CODE;
  AEHLT = strip(HLT_NAME);
  AEHLTCD = HLT_CODE;
  AEHLGT = strip(HLGTNAME);
  AEHLGTCDC = HLGTCD;
  AEBODSYS = strip(SOC_NAME);
  AEBDSYCD = SOC_CODE;
  AESOC = strip(SOC_NAME);
  AESOCCD = SOC_CODE;
  AESEV = upcase(strip(put(AESEVMAX, $AESEV)));
  AESER_SDTM = upcase(strip(put(AESER, $SNY)));
  AEACN_SDTM = ifc(AEC01AC ne ' ', 'MULTIPLE', ifc(AEACN = '4', 'DRUG WITHDRAWN', upcase(strip(put(AEACN, $AEACN)))));
  AEREL_SDTM = ifc(AEREL = 'OUT9488', 'REASONABLE POSSIBILITY AE CAUSED BY IP', ifc(AEREL = 'OUT9487', 'UNLIKELY AE CAUSED BY IP', ''));
  AEOUT_SDTM = upcase(strip(put(AEOUT, $OUT)));
  AECONTRT_SDTM = strip(put(AECONTRT, $SNY));
  %dtc(INVAR_DAT=AESTDAT,INVAR_TIM=AEESTIM,OUTVAR_DTC=AESTDTC);
  %dtc(INVAR_DAT=AEENDAT,INVAR_TIM=AEENTIM,OUTVAR_DTC=AEENDTC);
  AEENRTPT = ifc(AEENDAT = '' and AEOUT IN ('OUT9498','OUT9495'), 'BEFORE', ifc(AEOUT = 'OUT8275', 'COINCIDENT', ifc(AEOUT IN ('OUT9496','OUT9494'), 'ONGOING', '')));
  AEENTPT = ifc(AEENDAT ne ' ', ' ', ifc(AEOUT IN ('OUT9498','OUT9496','OUT9495','OUT9494'), 'STUDY TERMINATION', ifc(AEOUT = 'OUT8275', 'DEATH', '')));
  if AETERM = '' then delete;
run;
data AE_AE;
  set AE_AE_TMP(rename=(AEACN=AEACN_RAW AECONTRT=AECONTRT_RAW AEOUT=AEOUT_RAW AEREL=AEREL_RAW AESER=AESEV_RAW AETERM=AETERM_RAW));
  rename AEACN_SDTM=AEACN AECONTRT_SDTM=AECONTRT AEOUT_SDTM=AEOUT AEREL_SDTM=AEREL AESER_SDTM=AESEV AETERM_SDTM=AETERM;
run;
proc delete data=AE_AE_TMP; run;

/* DATASET: AE_SERAE */
data AE_SERAE_TMP;
  length AESCONG_SDTM AESDISAB_SDTM AESDTH_SDTM AESHOSP_SDTM AESLIFE_SDTM AESMIE_SDTM AESPID SAEDTC SAEHODTC SAEIADTC $200;
  set RAW.SERAE;
  STUDYID = strip(STUDY);
  USUBJID = CATS(STUDY,"/",SUBJECT);

proc sort data=AE_AE; by usubjid aeno; run;
proc sort data=AE_SERAE; by usubjid aeno; run;
data AE;
  merge AE_AE AE_SERAE;
  by usubjid aeno;
run;

/****** END OF AUTO PROGRAM *****/
```

Case Studies: Generated SAS Code for VS

```
***** START OF AUTO PROGRAM *****/  
  
/* DATASET: VS_VS */  
data VS_VS;  
  length VISITDTC VSDTC VSORRES VSORRESU VSSTRESC VSSTRESU VSTESTCD VSTPT $200;  
  set RAW.VS;  
  STUDYID = strip(STUDY);  
  DOMAIN = "VS";  
  USUBJID = CATS(STUDY,"/",SUBJECT);  
  VISITNUM = VISIT;  
  %dtc(INVAR_DAT=VSDAT,INVAR_TIM=VSTIM,OUTVAR_DTC=VSDTC);  
  VSTPT = strip(PROTSCHD);  
  MODULE = ifc(MODULE ne '',MODULE, 'VS');  
  %dtc(INVAR_DAT=VIS_DAT,OUTVAR_DTC=VISITDTC);  
  IF not missing(SBP) THEN DO;  
    VSTESTCD = "SYSBP";  
    VSTEST = "Systolic Blood Pressure";  
    VSCAT = "PULSE AND BLOOD PRESSURE";  
    VSPOS = "UNSPECIFIED";  
    VSORRES = strip(put(SBP, best.));  
    VSORRESU = "mmHg";  
    VSSTRESC = strip(put(SBP, best.));  
    VSSTRESN = SBP;  
    VSSTRESU = "mmHg";  
    OUTPUT;  
  END;  
  call missing(VSTEST, VSCAT, VSPOS, VSORRES, VSORRESU, VSSTRESC, VSSTRESN, VSSTRESU);  
  
  IF not missing(DBP) THEN DO;  
    VSTESTCD = "DIABP";  
    VSTEST = "Diastolic Blood Pressure";  
    VSCAT = "PULSE AND BLOOD PRESSURE";  
    VSPOS = "UNSPECIFIED";  
    VSORRES = strip(put(DBP, best.));  
    VSORRESU = "mmHg";  
    VSSTRESC = strip(put(DBP, best.));  
    VSSTRESU = "mmHg";  
    OUTPUT;  
  END;  
  call missing(VSTEST, VSCAT, VSPOS, VSORRES, VSORRESU, VSSTRESC, VSSTRESU);  
  
  IF not missing(WEIGHT) THEN DO;  
    VSTESTCD = "WEIGHT";  
    VSTEST = "WEIGHT";  
    VSCAT = "BODY MEASUREMENT";  
    VSORRES = strip(put(WEIGHT, best.));  
    VSORRESU = strip(put(WEIGHTU, $VSRES2F.));
```

Future Directions

- **Define.xml Tool Integration**

Plan to incorporate define.xml generation tools to streamline regulatory document creation.

- **aCRF Generation Integration**

Integrate aCRF generation tools to automate clinical case report form creation.

- **AI-based Mapping Enhancements**

Enhance AI algorithms to improve accuracy and efficiency in data mapping processes.

- **Incorporating AI Models**

Incorporate AI models or agents to automate decision-making and data processing tasks.

Conclusion

- **More General Automation of SDTM Repetitive Tasks**

The software automates repetitive mapping tasks, significantly improving efficiency and reducing manual errors.

- **Support for Complex Standards**

It supports complex clinical data standards, ensuring accurate and compliant SDTM mapping.

- **Customization and Future Integration**

Offers customization options and plans for integrating AI and regulatory tools for enhanced functionality.

A collection of geometric shapes on the left side of the slide: an orange outline of an upward-pointing triangle at the top, a teal solid downward-pointing triangle below it, and a purple outline of an upward-pointing triangle at the bottom.

THANK YOU

A collection of geometric shapes on the right side of the slide: an orange solid downward-pointing triangle at the top, a teal outline of an upward-pointing triangle below it, and a white outline of a downward-pointing triangle at the bottom.