

Safety ADaM Automation from Design to Implementation

Chengxin Li, Tingting Tian, Toshio Kimura
Regeneron Pharmaceuticals Inc.

Presenter: Chengxin Li

May 2022

REGENERON[®]

Agenda



Introduction



Method



Design and Implementation



Questions

Agenda



Introduction



Method



Design and Implementation



Questions

Introduction

	Background	❖ CDISC widely deployed in the past decade and becoming programming routine work ❖ Compliance still as focus → shifting towards reusable automated solution ❖ Programming automation scope autoACRF, autoSDTM, autoADaM, and autoTLF
	Safety autoADaM as good starting point	❖ Safety analyses more straightforward and based on actual observed data, less challenges compared with efficacy ❖ Single SDTM domain as input dataset
	Safety autoADaM implemented	❖ ADAE (OCCDS), ADLB(BDS), ADEG(BDS), and ADVS(BDS) ❖ ADSL also included due to the same method applied
	Limitation	❖ Currently limited to parallel study design supporting one or multiple treatment periods, easily extended to cross-over study

Agenda



Introduction



Method



Design and Implementation



Questions

Method: GPS Navigation

Global

- ❖ Global variables
 - ✓ Applicable to all projects and studies
 - ✓ Populated or derived by default without user input
 - ✓ Approximately 80% are global, fitting into 80/20 rule
 - ✓ Examples: RANDFL, SAFFL



Project

- ❖ Project variables
 - ✓ Applicable to all studies under the project
 - ✓ Dynamic depending on SAP
 - ✓ Controlled with macro parameters though user input
 - ✓ Examples: ASTDT/ASTDTF with imputations



Study

- ❖ Study variables
 - ✓ Applicable to the specific study
 - ✓ Dynamic depending on SAP
 - ✓ Controlled with macro parameters though user input
 - ✓ Examples: ADURN with end date imputations



Method: Sequential Modular Programing

	1	Initiation	1) Set system options 2) Define sub-macros 3) Build dataset templates 4) Record initiation status (used during cleaning-up)
	2	Parameter and Dataset Checking	1) Check LIBREFs, datasets, formats, or additional requirements such as on EPOCH 2) Check user input against defined rules and requirements 3) Set defaults (unless specified otherwise) 4) Prepare for derivations, e.g., parse parameters so that they are derivation-ready 5) Write outcomes of checks to SAS LOG window and Execution Summary Report
	3	Derivation	1) Perform variable derivations 2) Prepare the dataset and associated documents for output-ready
	4	Output	1) Store ADaM dataset and associated documents in the designated folder
	5	Cleaning-up	1) Reset settings to the status prior to the macro call (such as deleting intermediate datasets)

Method: User Experience

- Statistical programmer as primary user and statistician as another key stakeholder
- Easy to understand and easy to use, targeting to low transition cost

 Platform		Multiple platforms are supported, currently supports both PC/EG SAS
 Parameter	Name	Straightforward and intuitive to understand
	Input	Explicit input or with a look-up table
	Rules or Conventions	Consistent input patterns across macros and/or parameters such as using “#” denoting multiple items. Macro to catch ERRORS (such as invalid input parameters) to avoid unpredictable SAS messages
 Performance		Optimize performance by using faster processing methods such as PROC SQL or PROC COPY. Use of slow procedures such as sorting will be limited to only when it is necessary.
  Output	SAS LOG	Saved as an external file and replayed back to the SAS LOG Window
	Format	Use consistent formats across all datasets, e.g., all date in YYMMDD10. format
	Datasets	ADaM datasets stored in SAS and XPT formats.
	Documentation	Specifications are dynamically produced based on the macro call instance. Specifications layout will facilitate the use of define-xml tool.
	Facilitate Review	Execution Summary Report generated in separate PDF and Excel files

Agenda



Introduction



Method



Design and Implementation



Questions

Design and Implementation: %ADSL



Global variables derived in ADSL without user input

Variable Type	Common Variable	Source
All IDs	STUDYID, USUBJID, SUBJID, SITEID, INVID, ...	DM
Demographic	AGE, AGEU, SEX, RACE, ETHNIC, COUNTRY/CNTRYDC, ...	DM
DATE or DATETIME	RFICDT, RFSTDT, RFENDT, RANDDT, TRTSDT, TRTEDT, TRTSDTM, TRTEDTM, DTHDT, RFPENDT, TRxxSDT, TRxxSDTM, TRxxEDT, TRxxEDTM	DM, EX/EC, DS
Population Set	ENRFL, DTHFL, SAFFL, RANDFL, FASFL, COMPLEFL/COMPLTFL/ COMPLSFL/COMPxxFL (FASFL derived as Randomized and Treated)	DM, EX, DS
Treatment	ARM, ARMCD, ACTARM, ACTARMCD, TRTxxP, TRTxxPN, TRTxxA, TRTxxAN	DM, TA, EX
Disposition	DCEREAS/DCEREASP, DCTREAS/DCSREAS/DSTxxRS, DCSREASP/DCTREASP/DCTxxRSP	DS

Design and Implementation: %ADSL



Baseline: commonly, --TESTCD as baseline variable name, --TEST plus --STRESU as label, --STRESN copied as values



Other case derivations incorporated for baseline

Case	Derivation	Note	Example
Duplicate tests	Seq number will be added to the variable name; values of --POS or --CAT added to variable label	Currently only supporting variable --POS or --CAT but extensible	Two GLUC tests with CHEMISTRY and URINALYSIS differentiated by LBCAT
All values with NULL in populated variable --STRESN and/or --STRESC per test	1. If both --STRESN and --STRESC are all missing, then use --ORRES as baseline		
	2. If --STRESN are all missing but --STRESC is populated, then use --STRESC as baseline		Test results with POSITIVE or NEGATIVE
Code and decode per test	Two variables derived, one for code, another for decode: --STRESN as code, --ORRES or --STRESC as decode	Two mapping options supported: 1. decode mapped with --ORRES, code with --STRESC and --STRESN; 2. decode mapped with --ORRES and --STRESC; code mapped with --STRESN	Ordinal data like pain scales in QS
LOQ	--STRESC/--STRESN not all null but not matching, then copying --STRESN (NULL)	Currently a log WARNING issued with information DOMAIN, TESTCD, and USUBJID	BLD or BLQ, e.g., <10

Design and Implementation: %ADAE



Limitations, not supporting the following analyses

- ✗ AE of special interests(AESI)
- ✗ SMQ/CMQ analyses
- ✗ Exposure adjusted AE analyses



Global variables

- ✓ Most SDTM AE variables
- ✓ Treatment date/datetime variables
- ✓ Treatment group variables including derived TRTA
- ✓ Ordering variables for ordinal variables, e.g., AESEV, AEREL
- ✓ Two flag variables AOCCPIFL (1st Max Sev./Int. Occur Within PT Flag) and AOCCSIFL (1st Max Sev./Int. Occur Within SOC Flag)

Design and implementation: %ADAE



P/S variables
(Macro parameters)

Parameter	Core	Definition
POPU	Optional	Analysis population set variable to be merged from ADSL onto ADAE Default: SAFFL
ADSL_VAR	Optional	Supportive/core variables from ADSL that will be merged onto ADAE Default: SITEID
STDT_IMPUT	Optional	Enable imputation of the AE onset date when the AE onset date is missing
ENDT_IMPUT	Optional	Select the imputation method for AE end date when the AE end date is missing. In order to derive ADURN, the AE end date imputation method must be defined. Default: AE end date imputation method is not defined; thus, ADURN is not derived
ATTR_IMPUT	Optional	AE attribute variable imputation list, e.g.,
GROUPING	Optional	Grouping variable definition using a SAS format
LAG	Optional	Define the number of days used as the lag for the treatment phase If a value of 0 is entered, then there will be no lag. If "MAX" is entered, then the time until the next treatment period will be considered as part of the current treatment period, and follow-up phase considered as last treatment period. Default: 0
SUPPQ	Optional	List the QNAMs to be merged into ADAE from SUPPAE

Design and Implementation: %ADLB



P/S variables
(Macro parameters)

Parameter	Definition
LOQ	Value used when limit of quantification (LOQ) has been reached
POPU	Analysis population set variable to be merged from ADSL onto ADLB
ADSL_VAR	Supportive/core variables from ADSL that will be merged onto ADLB
AVISIT	Visit mapping or timing windows are derived by applying a format to a Study Day variable such as ADY, LBDY, VISITDY. If nothing is specified, the value for VISIT will be used as the value for AVISIT.
AVISITN	Numeric ordering variable associated with AVISIT will be derived using a SAS informat. If nothing is specified, the value for VISITNUM will be used as the value for AVISITN.
AWTARGET	AWTARGET variable will be derived using a SAS informat on AVISIT or ADY. If nothing is specified, the value for VISITDY or retaining VISITDY of parent visits for unscheduled visits will be used as the value for AWTARGET
ANALFL	Data selection method for ANL01FL when there are multiple records within a single AVISIT Options: LAST (default), FIRST, NOMINAL (closest to the target date; if tie then selects the earliest)
LAG	Define the number of days used as the lag that will be used to derive APHASE. If a value of 0 is entered, then there will be no lag. If "MAX" is entered, then the time until the next treatment period will be considered as part of the current treatment period, and follow-up phase considered as last treatment period.
SHIFT	Parameter is used to define the derivation for the SHIFT variable Default: SHIFT1 is derived using BNRIND and ANRIND
SHIFTN	Numeric ordering variable associated with SHIFT will be derived using a SAS informat.
ADDCRIT	Parameter is used to derive the CRITy/CRITyFL variables. Only a single criterion can be used.
ADDPARAM	Parameter is used to create a new PARAM for a compound criteria derivation or provide an alternative method of ADDCRIT which is limited to a single criterion.
GROUPING	Derive the CAT variable using a SAS format or specify a dataset
ORDERING	Derive numeric sorting variable for the CAT variable derived through the GROUPING parameter by applying a SAS informat.
SUPPQ	List the QNAM's to be merged into ADLB from SUPPLB
LBCOM	List LB related IDVARs to derive Lab Comments from SDTM CO.COVAL; currently the support limited to parent record(s) level

Design and Implementation: Example Call

```
%ADVS(  
  SDTMLIB =%str(SDTM),           /*provide source SDTM dataset libname*/  
  ADaMLIB =%str(ADaM),           /*provide output dataset libname, output files: SAS/XPT, LOG and SPECS, Execution Summary Report*/  
  MISCLIB =%str(MISC),           /*provid misc dataset libname*/  
  
  POPU    =%str(SAFFL),           /*provide analysis population set variable, SAFFL by default, # separated if multiple*/  
  ADSL_VAR=%str(RACE # SEX),      /*provide supportive/core variables from ADSL, # separated if multiple*/  
  
  AVISIT  =%str(VISIT/format=$AVISIT.), /*remapping VISIT or applying timing window, e.g., with ADY/format=$AVISIT. By default, copying VISIT*/  
  AVISITN =%str(AVISIT/informat=AVISITN.), /*ordering AVISIT. By default, copying VISITNUM*/  
  AWTARGET=%str(AVISIT/informat=AVISITN.), /*deriving Analysis Window Target with AVISIT/informat=xxx or ADY/informat=yyy*/  
  ANALFL  =%str(AVISIT=FIRST/ATPT=AVG), /*selecting method for ANL01FL derivation of AVISIT, option: LAST, FIRST, NOMINAL(WORST not done yet)*/  
                                           /*for ADVS and ADEG, option ATPT=AVG/MIN/MAX enabled with ANL02FL, new record added and DTYPE assigned*/  
  ADDCRIT =,                      /*CRITy/CRITyFL derived in the same row*/  
  ADDPARAM=,                      /*deriving new parameter for either compound criteria or single criteria*/  
  
  GROUPING=,                      /*deriving CAT variable by user-defined format*/  
  ORDERING=,                      /*ordering variable by user-defined informat from*/  
  
  SUPPQ =,                        /*list of QNAMs from SUPPLB derived to ADLB*/  
  SPECS  =%str(Y)                 /*generating ADLB specs in EXCEL format and saved in ADaMLIB*/  
);
```

Summary



- Safety-based autoADaM successfully implemented with SAS macros
 - ❖ GPS navigation method
 - ❖ Sequential structured programming method
 - ❖ UI and UX design to facilitate use



- Next Steps
 - ❖ Parameters to be added for plain SAS code generation
 - ✓ Facilitate review
 - ✓ Resolve submission challenge
 - ❖ Develop Graphical User Interface to further simplify use

Agenda



Introduction



Method



Design and Implementation



Questions

Questions

Chengxin Li
Associate Director
Regeneron Pharmaceutical Inc.
Tarrytown, NY
chengxin.li@regeneron.com

