PP21

ADaM programming made easy – Common ADaM templates and a SAS macro library

Niels-Kristian Kjøller¹; Parag Wani²; Charlotte Bergknut²; Ari Knoph³; Jeppe Juul²; Anne Katrine Alsing²; Jesper Zeth³

¹Novo Nordisk A/S, Søborg, Denmark (nkrk@novonordisk.com); ²Novo Nordisk A/S, Søborg, Denmark; ³Novo Nordisk A/S, Aalborg, Denmark

Goal

Jumpstart ADaM programming in a structured and modular approach using:

- ADaM template programs for Basic Data Structure (BDS) and Occurrence Data Structure (OCCDS) datasets
- A library of data derivation and utility components

Motivation

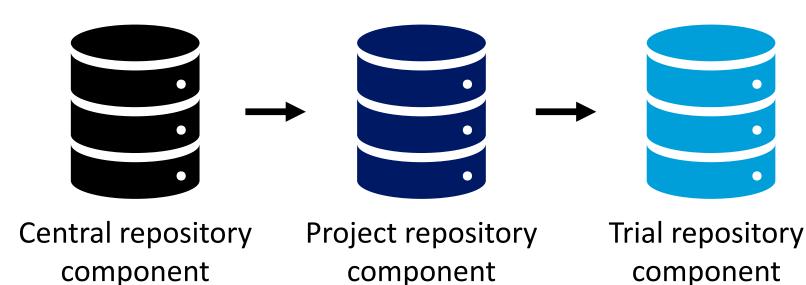
- 1. Ease of use:
- Saves time in development and maintenance
- More time for trial specific needs
- 2. Data readiness:
 - 80% of desired ADaM variables ready as soon as SDTM is ready
- 3. Quality:
- Same code used for the same derivations across different datasets, trials, and projects
- Ensures good programming practice

Components

Our approach is built around a central repository of data derivation components. The components can be either macros or code snippets. They cover derivations like date/time imputation, retest rules, analysis flags, limit of quantification and screening observation carried forward.

The ADaM template programs are built using the data derivation components as building blocks. Each component can be replaced with another component according to project or trial specific requirements.

Figure 1 The data derivation component hierarchy



Data derivation components exist at three levels; company, project and trial level. Likewise, the templates exist at a company level, but are adapted to project needs within each project, which again form the basis or starting point for each trial.

Metadata

The ADaM template programs rely on metadata from several sources. Following are some of the key types of metadata:

- Visit metadata (Labels for each visit)
- Parameter metadata (Labels and units for each parameter code)
- Flowchart metadata (Planned assessments at each visit)
- Formatting metadata (Format, length, and label for each variable in final ADaM dataset)

ADaM template programs

The structure and program flow of the BDS/OCCDS templates are visualized in Figure 2. The functional steps have been grouped into code blocks for ease of readability, with the OCCDS flow having only a subset of the blocks of the BDS flow.

Figure 2 Program flow of the ADaM template programs

Basic Data Structure program flow DAT: Retrieve datasets %access Create libraries Extract SDTM Merge xx and SUPPxx, rename variables, create traceability variables Select data in scope, pre-allocate analysis flags and DTYPE, create analysis visit AVISITN, attach variables from ADSL DTM: Create date/time variables Impute date – visit date Perform date imputation and create date variables ADT, ADTF Create analysis time ATM, relative day ADY ELI: Flag records eligible for analysis

Mark eligible records, i.e. remove eligibility flag for unscheduled assesements using metadata [flowchart metadata], etc.

Retest value – first value

Apply retest rule to remove elibility flag for repeated measurements

Flag baseline records using metadata [flowchart metadata]

Screening observation carried forward

PER: Define analysis flags based on observation periods

If baseline missing, carry forward screening observation

In-trial period

Mark records belonging to the in-trial period

On-treatment period

Mark records belonging to the on-treatment period

VAL: Creating the analysis value

Analysis values AVAL and AVALC are created from SDTM, clinical significance is appended (if applicable)

Upper limit of quantification

Apply LOQ rule for assesements above upper limits

Lower limit of quantification

Apply LOQ rule for assesements below lower limits

DER: Perform additional data derivations

Most dataset specific derivations will appear here. Example:

Derive BMI records.

ABL: Create baseline variables

Baseline values, BASE, BASEC, CHG, PCHG, R2BASE are identified and added to all records

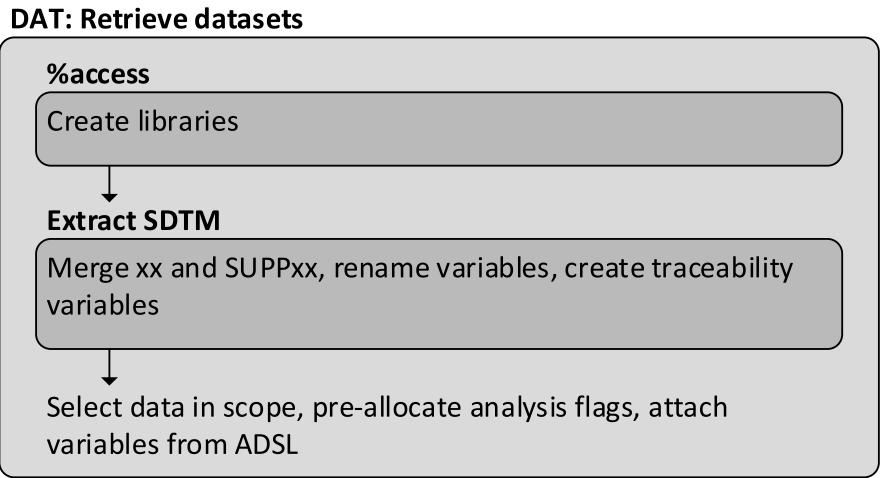
FIN: Finalize dataset

Visit labels AVISIT, parameter labels PARAM and analysis units AVALU are attached from metadata [visit metadata, parameter metadata]

→ %adam_finalizer

Sort data, assign sequence (ASEQ), apply formats, labels etc. from metadata [formating metadata], save final dataset

Occurrence Data Structure program flow



DTM: Create date/time variables

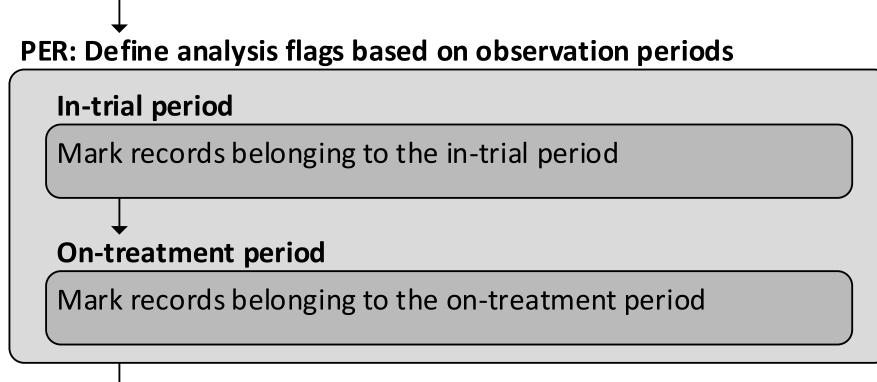
Impute start date – early

Perform date imputation and create start date variables
ASTDT, ASTDTF

Impute end date – late

Perform date imputation and create end date variables
AENDT, AENDTF

Create analysis start/end time ASTTM/AENTM, relative start/end day ASTDY/AENDY



OTH: Other derivations

Most dataset specific derivations will appear here. Example:
Assign standard MedDRA queries

FIN: Finalize dataset

%adam_finalizer

Sort data, assign sequence (ASEQ), apply formats, labels etc. from metadata [formatting metadata], save final dataset

Light grey boxes represent program code blocks.

Dark grey boxes represent components used within each block.

Blue text highlights metadata use.

In Figure 2, sub-text in each block contains examples of what is included in the templates, with the central repository components being the default. However, each block also serve as a placeholder for the substitution or addition of trial specific derivation such as e.g. additional analysis periods if needed.

Conclusion

- ADaM template programs can be utilized to optimize resources and to fast track data processing for clinical trials while ensuring quality and good programming practice
- Two example templates are presented here one for Basic Data Structure (BDS) datasets and one for Occurrence Data Structure (OCCDS) datasets
- The template programs ensure that the same code is used for the same derivations across different datasets, trials, and projects