Paper PP21

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

Niels-Kristian Kjøller, Novo Nordisk A/S, Søborg, Denmark
Parag Wani, Novo Nordisk A/S, Søborg, Denmark
Charlotte Bergknut, Novo Nordisk A/S, Søborg, Denmark
Ari Knoph, Novo Nordisk A/S, Aalborg, Denmark
Jeppe Juul, Novo Nordisk A/S, Søborg, Denmark
Anne Katrine Alsing, Novo Nordisk A/S, Søborg, Denmark
Jesper Zeth, Novo Nordisk A/S, Aalborg, Denmark

ABSTRACT

In this paper we present two ADaM program templates. One common template for all Basic Data Structure programs (e.g. ADLB), and one for all Occurrence Data Structure programs (e.g. ADAE). The templates are built around short and easy-to-read macros. The structure is modular, and each macro can easily be swapped out to suit trial needs. This approach saves time and ensures aligned definitions (both across datasets within a trial and across trials within a project) as well as ensuring GPP. It allows 80% of desired variables to be ready as soon as SDTM is ready and enables the programmer to focus on trial specific details.

INTRODUCTION

Within an organization, there is often a desire to align how clinical trials define key concepts – whether it be a simple concept such as the definition of trial start/end date or more complicated matters like handling missing data. There are many potential benefits including time efficiency, ease of use, higher quality, re-usability and can translate down-stream to earlier access to analysis data. At the same time, it can facilitate a framework in which it is easier to enforce and ensure that GPP is applied.

With respect to programming, alignment means using the same code for the same derivations across different datasets, trials and projects. However, trials are different, and the same definitions do not necessarily apply across trials. Within the organization, different projects might cover different therapeutic areas, and different trials have different focus. As the saying goes “one size fits none” and a rigid model will not cater to all trials and projects.

In this paper we present an approach that seeks to balance the desire for consistent definitions and reuse of code with the flexibility to suit individual project and trial related needs. Our model consists of a repository of data derivation components and a set of ADaM program templates built around these components. As an example of this model, we present how we structure ADaM template programs for Basic Data Structure (BDS) and Occurrence Data Structure (OCCDS) datasets.

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.

ADaM template programs are built using the data derivation components as building blocks. The modular nature of the template programs makes it possible to replace a component with another component according to project or trial specific requirements.

Data derivation components exist at three levels; company, project and trial level. See Figure 1. 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.

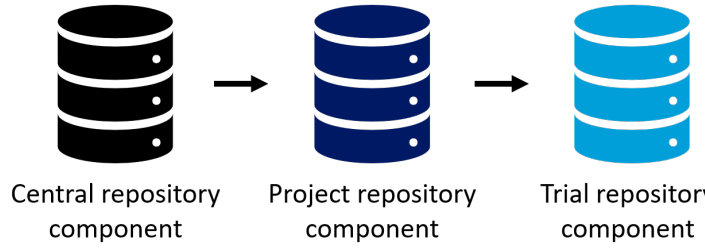

Figure 1 The data derivation component hierarchy

METADATA

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

- Visit metadata
- Parameter metadata
- Flowchart metadata
- Formatting metadata

Visit metadata contain visit labels corresponding to each value of AVISITN. Parameter metadata contain labels and units corresponding to each value of PARAMCD. Flowchart metadata list all planned assessments by PARAMCD and AVISITN, and contain information on assessment type (baseline, end-of-treatment, etc.). Formatting metadata contain information on how the final ADaM dataset is to be presented, including format, length, label and variable order for each variable as well as sorting keys for each dataset. Other metadata datasets could be created for of standard MedDRA queries for adverse events, Anatomical Therapeutic Chemical (ATC) classification codes for medical history, normal ranges for lab measurements, etc.

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.

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. Within each block, repository components are represented by a dark grey box. However, each block also serves as a placeholder for the substitution or addition of trial specific derivation such as e.g. additional analysis periods if needed.

Below, the function of each block is briefly outlined. Some sections are marked with (BDS) or (OCCDS) to reflect that not all sections are relevant for both BDS and OCCDS templates.

DAT: RETRIEVE DATASETS

Source libraries are accessed, and the relevant SDTM domain is merged with the corresponding supplementary dataset SUPP--. The data in scope for analysis is selected and simple renaming is performed including the creation of traceability variables --SEQ (for OCCDS templates) or SRC--- (for BDS templates). Then subject level variables are attached from ADSL.

DTYPE and analysis flags are pre-allocated for later use. Three analysis flags are assigned: The eligibility flag marking which records are in scope for analysis, the in-trial flag marking records in the observation period and the on-treatment flag marking records in the exposure period. By default, all records are marked as eligible at this stage.

DTM: CREATE DATE/TIME VARIABLES

Date and time are converted from ISO format to SAS format and relative analysis day is derived. Missing or partial dates are imputed using one of the several imputation rule components from the repository. Time is not imputed. For BDS datasets the following variables are created; ATM, ADT, ADTF and ADY. For OCCDS datasets the corresponding start/end date variables are created; ASTTM, AENTM, ASTDT, AENDT, ASTDTF, AENDTF, ASTDY and AENDY.

Figure 2 Program flow of the ADaM template programs

Light grey boxes represent program code blocks.
Dark grey boxes represent components used within each block.
Blue text highlights metadata use.

ELI: FLAG RECORDS ELIGIBLE FOR ANALYSIS (BDS)
Unscheduled assessments and other assessments violating pre-specified criteria are marked as ineligible for analysis. Flowchart metadata is used to identify unscheduled assessments. For laboratory data, retest rules are applied to mark repeated measurements as ineligible.

BAS: FLAG BASELINE RECORDS (BDS)
Using planned baseline information from the flowchart metadata dataset, the baseline records are identified for each type of assessment. If baseline observations are missing, screening observations are carried forward.

PER: DEFINE ANALYSIS FLAGS BASED ON OBSERVATION PERIODS
The in-trial flag, marking records in the observation period, and the on-treatment flag, marking records in the exposure period (including lag time) are marked.

VAL: CREATING THE ANALYSIS VALUE (BDS)
Analysis value AVAL/AVALC is created from SDTM. For AVALC, clinical significance is appended if applicable. For laboratory data, limit of quantification (LOQ) rules are applied to handle values above/below the upper/lower limits of quantification.

DER: PERFORM ADDITIONAL DATA DERIVATIONS (BDS)
In this block, the program is further customized to fit dataset, trial or project specific needs. Most dataset specific derivations will appear here. An example is the derivation of new records for BMI, based on height and weight records.

ABL: CREATE BASELINE VARIABLES (BDS)
Having derived all new records in the DER block, the baseline value variable BASE and the variables relative to baseline such as CHG, PCHG and R2BASE are derived for all records.

OTH: OTHER DERIVATIONS (OCCDS)
Additional dataset specific derivations can be performed here to further customize the program. An example is the assignment of standard MedDRA queries in ADAE, which is done using a separate MedDRA metadata dataset.

FIN: FINALIZE DATASET
For BDS programs, visit level metadata, including AVISIT, and parameter level metadata, including PARAM and AVALU, are attached in this step. The dataset is sorted, assigned a unique sequence number ASEQ and saved with correct formats, lengths, labels and variable order according to specifications from formatting metadata.

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.

Key to this approach is a repository of data derivation components. Data derivation components exist at three levels; company, project and trial level. The templates are governed at company level but are adapted to project needs within each project, which again form the starting point for each trial.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Niels-Kristian Kjøller
Novo Nordisk A/S
Vandtårnsvej 114
DK-2860 Søborg
nkrk@novonordisk.com
www.novonordisk.com