Purpose
CDISC has laid strict Analysis Data Model Implementation Guideline (ADaMIG), specifying ADaM standard dataset structures and variables, including naming conventions, labels, types, formats etc. A programming solution is required to effectively adhere to the standard without the risk of making mistakes and duplication of work. This poster describes a method which programmatically uses the ADaM specification to shape the structure of the final ADaM datasets.

Challenges
- Assigning variable attributes
- Implementing changes
- Maintaining standards
- Governing sequence of the variables

Solution
This ADaM standardization technique is based on two utilities which are used in pairs.

1. Utility 1
Reads in the ADaM specification document and creates 2 types of datasets:
- Analysis variable metadata for all domains available in the specification
- Empty SAS datasets (Skeletons) with the same structure as the final ADaM dataset for each domain

2. Utility 2
- Appends last Work.dataset created in the program to the ADaM skeleton dataset (created by Utility 1) for the specified domain
- Sorts the data according to sorting key available in the Analysis variable metadata (created by Utility 1)
- Creates a discrepancy report

Conclusions
- Programmatically using the ADaM specification
- No need to remember which variables to keep or drop
- Ease of governing the sequence of variables
- Ease of maintaining the standards
- All the variables and their attributes are listed in SAS dataset containing the Analysis variable metadata
- Easy to check for inconsistencies and make necessary corrections
- Can be used by the sponsor when checking for structural inconsistencies in the ADaM datasets delivered by a CRO