

# Creating a basic data structure (BDS) Exposure ADaM

Lun-Hsien Chang

2024-12-01

## Programming workflow

<b>Read CDISC pilot SDTM and ADaM datasets</b>	<b>2</b>
EXADJ . . . . .	2
EXDOSE . . . . .	2
EXPLDOS . . . . .	2
<b>Derive numeric datetime, analysis day variables</b>	<b>3</b>
ASTDT . . . . .	3
AENDT . . . . .	3
ASTDTM . . . . .	4
AENDTM . . . . .	4
ASTDY . . . . .	5
AENDY . . . . .	5
EXDURD . . . . .	5
DOSEO . . . . .	6
PDOSEO . . . . .	6
Create 1:1 mapping records . . . . .	6
PARAMCD . . . . .	7
AVALC . . . . .	7
Create Summary Records . . . . .	8
Create Summary Records . . . . .	8
Assign PARAMCD, PARAMN, etc. from Reference tables . . . . .	11
<b>Derive Categorization Variables (AVALCATx)</b>	<b>12</b>
AVALCAT1 . . . . .	12
Assign ASEQ . . . . .	12
ASEQ . . . . .	12
Add Labels and Attributes . . . . .	13
<b>References</b>	<b>13</b>
Get required R packages	

Warning: package 'pharmaversesdtm' was built under R version 4.4.2

Attaching package: 'lubridate'

The following objects are masked from 'package:base':

date, intersect, setdiff, union

## Read CDISC pilot SDTM and ADaM datasets

```
adsl_vars <- exprs(TRTSDT, TRTSDTM, TRTEDT, TRTEDTM)

# left join EX and adsl TRTSDT, TRTSDTM, TRTEDT, TRTEDTM on ex.STUDYID=adslSTUDYID and ex.USUBJID
adex <- derive_vars_merged(
  dataset=ex
  ,dataset_add = adsl
  ,new_vars = adsl_vars
  ,by_vars = exprs(STUDYID, USUBJID)
) # dim(adex) 591 21
```

The CDISC pilot EX domain data does not contain a dose adjustment flag or the planned dose information. For demonstration purposes, this will be added to the data.

### EXADJ

- Exposure Adjustment ?

### EXDOSE

- exposure dose
- from SDTM.EX.EXDOSE

### EXPLDOS

- Planned Dose

```

adex <- adex %>%
  mutate(
    EXADJ = case_when(
      USUBJID == "01-701-1028" & VISIT %in% c("WEEK 2") ~ "ADVERSE EVENT",
      USUBJID == "01-701-1148" & VISIT %in% c("WEEK 2", "WEEK 24") ~ "MEDICATION ERROR",
      TRUE ~ NA_character_
    ),
    EXDOSE = case_when(
      USUBJID == "01-701-1028" & VISIT %in% c("WEEK 2") ~ 0,
      USUBJID == "01-701-1148" & VISIT %in% c("WEEK 2", "WEEK 24") ~ 0,
      TRUE ~ EXDOSE
    )
  ) %>%
  mutate(EXPLDOS = if_else(EXTRT == "PLACEBO", 0, 54))

adex %>% select(EXTRT, EXPLDOS) %>% distinct()

```

```

# A tibble: 2 x 2
  EXTRT      EXPLDOS
  <chr>      <dbl>
1 PLACEBO          0
2 XANOMELINE      54

```

## Derive numeric datetime, analysis day variables

### ASTDT

- Analysis Start Date
- Set to a numeric form of EX.EXSTDTC when EX.EXSTDTC consists of a full date.

### AENDT

- Analysis End Date
- Set to a numeric form of EX.EXENDTC when EX.EXENDTC consists of a full date.

```

# Convert character datetime to numeric datetime
adex <- derive_vars_dt(adex, new_vars_prefix = "AST", dtc = EXSTDTC)
adex <- derive_vars_dt(adex, new_vars_prefix = "AEN", dtc = EXENDTC) # dim(adex) 591 25
adex %>% select(USUBJID, VISIT, EXSTDTC, EXENDTC, ASTDT, AENDT) %>% head()

```

```
# A tibble: 6 x 6
  USUBJID VISIT EXSTDTC EXENDTC ASTDT AENDT
  <chr>    <chr>   <chr>    <chr>    <date> <date>
1 01-701-1015 BASELINE 2014-01-02 2014-01-16 2014-01-02 2014-01-16
2 01-701-1015 WEEK 2    2014-01-17 2014-06-18 2014-01-17 2014-06-18
3 01-701-1015 WEEK 24   2014-06-19 2014-07-02 2014-06-19 2014-07-02
4 01-701-1023 BASELINE 2012-08-05 2012-08-27 2012-08-05 2012-08-27
5 01-701-1023 WEEK 2    2012-08-28 2012-09-01 2012-08-28 2012-09-01
6 01-701-1028 BASELINE 2013-07-19 2013-08-01 2013-07-19 2013-08-01
```

## ASTDTM

## AENDTM

The next examples demonstrates the datetime imputation features available in the `derive_vars_dtm()` function, where the time is imputed as “00:00:00”:

```
adex <- derive_vars_dtm(
  adex
  ,dtc = EXSTDTC
  # Impute dtc date to the first day of the month
  ,highest_imputation = "M"
  ,date_imputation = "first"
  ,new_vars_prefix = "AST"
)

adex <- derive_vars_dtm(
  adex,
  dtc = EXENDTC,
  # Impute dtc date to the last day of the month
  highest_imputation = "M",
  date_imputation = "last",
  new_vars_prefix = "AEN"
)

adex %>% select(EXSTDTC,EXENDTC,ASTDTM,AENDTM) %>% head()
```

```
# A tibble: 6 x 4
  EXSTDTC EXENDTC ASTDTM AENDTM
  <chr>    <chr>    <dtm>    <dtm>
1 2014-01-02 2014-01-16 2014-01-02 00:00:00 2014-01-16 00:00:00
2 2014-01-17 2014-06-18 2014-01-17 00:00:00 2014-06-18 00:00:00
3 2014-06-19 2014-07-02 2014-06-19 00:00:00 2014-07-02 00:00:00
4 2012-08-05 2012-08-27 2012-08-05 00:00:00 2012-08-27 00:00:00
5 2012-08-28 2012-09-01 2012-08-28 00:00:00 2012-09-01 00:00:00
6 2013-07-19 2013-08-01 2013-07-19 00:00:00 2013-08-01 00:00:00
```

## ASTDY

- Analysis Start Day
- 'ASTDT-TRTSDT+1'

## AENDY

- Analysis End Day
- 'AENDT-TRTSDT+1'

```
adex <- derive_vars_dy(  
  dataset=adex  
  ,reference_date = TRTSDT  
  ,source_vars = exprs(ASTDT, AENDT)  
  ) # dim(adex) 591 33  
adex %>% select(TRTSDT, ASTDT, ASTDY, AENDT, AENDY) %>% head()
```

```
# A tibble: 6 x 5  
  TRTSDT      ASTDT      ASTDY AENDT      AENDY  
  <date>    <date>    <dbl> <date>    <dbl>  
1 2014-01-02 2014-01-02      1 2014-01-16     15  
2 2014-01-02 2014-01-17     16 2014-06-18    168  
3 2014-01-02 2014-06-19    169 2014-07-02    182  
4 2012-08-05 2012-08-05      1 2012-08-27     23  
5 2012-08-05 2012-08-28     24 2012-09-01     28  
6 2013-07-19 2013-07-19      1 2013-08-01     14
```

## EXDURD

- Duration of treatment or exposure
- 'EXDURD=AENDT - ASTDT +1'

```
adex <- adex %>%  
  derive_vars_duration(  
    new_var = EXDURD  
    ,start_date = ASTDT  
    ,end_date = AENDT  
    # duration unit can be "years", "months", "weeks", "days", "hours", "minutes", "seconds"  
    ,out_unit = "DAYS")  
adex %>% select(ASTDT, AENDT, EXDURD) %>% head()
```

```
# A tibble: 6 x 3
  ASTDT      AENDT      EXDURD
  <date>    <date>    <dbl>
1 2014-01-02 2014-01-16      15
2 2014-01-17 2014-06-18     153
3 2014-06-19 2014-07-02      14
4 2012-08-05 2012-08-27      23
5 2012-08-28 2012-09-01       5
6 2013-07-19 2013-08-01      14
```

## DOSEO

- Dose Overall (?). Refers to the actual dose of the study drug administered, standardized (e.g., per unit of body weight, such as mg/kg).
- 'EXDOSE \* EXDURD'

## PDOSEO

- Planned Dose Overall (?) Represents the planned dose per protocol, similarly normalized
- 'EXPLDOS \* EXDURD'

```
adex <- adex %>%
  mutate(
    DOSEO = EXDOSE * EXDURD
    ,PDOSEO = EXPLDOS * EXDURD)

adex %>% select(USUBJID, EXDOSE, EXPLDOS, EXDURD, DOSEO, PDOSEO) %>% head()
```

```
# A tibble: 6 x 6
  USUBJID      EXDOSE EXPLDOS EXDURD DOSEO PDOSEO
  <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl>
1 01-701-1015      0       0     15      0      0
2 01-701-1015      0       0    153      0      0
3 01-701-1015      0       0     14      0      0
4 01-701-1023      0       0     23      0      0
5 01-701-1023      0       0      5      0      0
6 01-701-1028     54      54     14    756    756
```

## Create 1:1 mapping records

The first set of exposure records to create will be records mapped 1:1 to an existing collected exposure record in SDTM. For these records, the AVAL or AVALC would be calculated using columns that exist on the data and no summarizing of records would be necessary.

These records may be used for input into summary records or be used individually for summarization in outputs. Some examples may be exposure duration, dose administered, dose adjusted, etc. based on one exposure record in SDTM.

These records can be derived using simple `dplyr::mutate` assignments and then combined

## PARAMCD

- Parameter Code

## AVALC

- Analysis Value Character ?

```
adex_durd <- adex %>%
  mutate(
    PARAMCD = "DURD",
    AVAL = EXDURD)

adex_dose <- adex %>%
  mutate(
    PARAMCD = "DOSE",
    AVAL = DOSEO)

adex_pldos <- adex %>%
  mutate(
    PARAMCD = "PLDOSE",
    AVAL = PDOSEO)

adex_adj <- adex %>%
  mutate(
    PARAMCD = "ADJ",
    AVALC = if_else(!is.na(EXADJ), "Y", NA_character_)
  )

adex_adjae <- adex %>%
  mutate(
    PARAMCD = "ADJAE",
    AVALC = if_else(EXADJ == "ADVERSE EVENT", "Y", NA_character_)
  )

adex <- bind_rows(
  adex_durd,
  adex_dose,
  adex_pldos,
  adex_adj,
```

```
adex_adjae) %>%
mutate(PARCAT1 = "INDIVIDUAL") # dim(adex) 2955 40

adex %>% select(USUBJID,VISIT,ASTDT,AENDT,PARAMCD,AVAL,AVALC) %>% head()
```

```
# A tibble: 6 x 7
  USUBJID VISIT ASTDT AENDT PARAMCD AVAL AVALC
  <chr>    <chr> <date> <date> <chr> <dbl> <chr>
1 01-701-1015 BASELINE 2014-01-02 2014-01-16 DURD 15 <NA>
2 01-701-1015 WEEK 2 2014-01-17 2014-06-18 DURD 153 <NA>
3 01-701-1015 WEEK 24 2014-06-19 2014-07-02 DURD 14 <NA>
4 01-701-1023 BASELINE 2012-08-05 2012-08-27 DURD 23 <NA>
5 01-701-1023 WEEK 2 2012-08-28 2012-09-01 DURD 5 <NA>
6 01-701-1028 BASELINE 2013-07-19 2013-08-01 DURD 14 <NA>
```

## Create Summary Records

Exposure is commonly analyzed by a timing interval (e.g. APHASE, APERIOD, AVISIT, etc.). For these types of calculations, the `derive_param_exposure()` function may be used. In addition to creating a summarized AVAL, the function will also compute minimum and maximum dates for the record. For example, to calculate the total dose by subject and treatment

```
adex <- derive_param_exposure(
  dataset = adex
  ,dataset_add = adex
  ,by_vars = exprs(STUDYID, USUBJID, !!!adsl_vars)
  ,input_code = "DOSE"
  ,set_values_to = exprs(
    PARAMCD = "TDOSE",
    PARCAT1 = "OVERALL",
    AVAL = sum(AVAL, na.rm = TRUE)
  )
) # dim(adex) 3209 40
```

## Create Summary Records

Exposure is commonly analyzed by a timing interval (e.g. APHASE, APERIOD, AVISIT, etc.). For these types of calculations, the `derive_param_exposure()` function may be used. In addition to creating a summarized AVAL, the function will also compute minimum and maximum dates for the record. For example, to calculate the total dose by subject and treatment

A record with `PARAMCD == "TDOSE"` is created with `PARCAT1` set to "OVERALL" using the records in ADEX where `PARAMCD == "DOSE"` by summing AVAL. In addition, the `ASTDT`, and `AENDT` are created as the minimum and maximum date/times associated with each `by_vars` grouping. Note that, in addition to `PARAMCD`, `PARCAT1`, `AVAL`, `ASTDT` and `AENDT`, only those variables specified in the `by_vars` argument will be populated in the new records.



```

adsl_vars <- exprs(TRTSDT, TRTSDTM, TRTEDT, TRTEDTM)

# Remove existing TDOSE records
adex <- adex %>% filter(PARAMCD != "TDOSE")

# Derive new records for the parameter TDOSE (Total Dose) by summarizing exposure data
adex <- derive_param_exposure(
  dataset = adex
  ,dataset_add = adex
  ,by_vars = exprs(STUDYID, USUBJID, !!!adsl_vars)
  # Specifies which parameter (DOSE) to summarize. This tells the function to use rows with PA
  ,input_code = "DOSE"
  ,set_values_to = exprs(
    PARAMCD = "TDOSE"
    ,PARCAT1 = "OVERALL"
    ,AVAL = sum(AVAL, na.rm = TRUE)
  )
) # dim(adex) 3209 40

```

Multiple parameters (records) may be created at one time using the `call_derivation()` function:

```
unique(adex$PARAMCD) # [1] "DURD" "DOSE" "PLDOSE" "ADJ" "ADJAE" "TDOSE"
```

```
[1] "DURD" "DOSE" "PLDOSE" "ADJ" "ADJAE" "TDOSE"
```

```

adex <- adex %>%
  filter(!PARAMCD %in% c("TDOSE", "TPDOSE", "TDURD", "TADJ", "TADJAE")) # dim(adex) 2955 40

# Derive new records for multiple parameters by summarizing exposure data
adex <- adex %>%
  call_derivation(
    derivation = derive_param_exposure,
    variable_params = list(
      params(
        set_values_to = exprs(
          PARAMCD = "TDOSE",
          PARCAT1 = "OVERALL",
          AVAL = sum(AVAL, na.rm = TRUE)
        ),
        input_code = "DOSE"
      ),
      params(
        set_values_to = exprs(
          PARAMCD = "TPDOSE",
          PARCAT1 = "OVERALL",
          AVAL = sum(AVAL, na.rm = TRUE)
        )
      )
    )
  )

```

```

    ),
    input_code = "PLDOSE"
  ),
  params(
    set_values_to = exprs(
      PARAMCD = "TDURD",
      PARCAT1 = "OVERALL",
      AVAL = sum(AVAL, na.rm = TRUE)
    ),
    input_code = "DURD"
  ),
  params(
    set_values_to = exprs(
      PARAMCD = "TADJ",
      PARCAT1 = "OVERALL",
      AVALC = if_else(sum(!is.na(AVALC)) > 0, "Y", NA_character_)
    ),
    input_code = "ADJ"
  ),
  params(
    set_values_to = exprs(
      PARAMCD = "TADJAE",
      PARCAT1 = "OVERALL",
      AVALC = if_else(sum(!is.na(AVALC)) > 0, "Y", NA_character_)
    ),
    input_code = "ADJAE"
  )
),
dataset_add = adex,
by_vars = exprs(STUDYID, USUBJID, !!!adsl_vars)
)

count(adex, PARAMCD, PARCAT1)

```

# A tibble: 10 x 3

	PARAMCD	PARCAT1	n
	<chr>	<chr>	<int>
1	ADJ	INDIVIDUAL	591
2	ADJAE	INDIVIDUAL	591
3	DOSE	INDIVIDUAL	591
4	DURD	INDIVIDUAL	591
5	PLDOSE	INDIVIDUAL	591
6	TADJ	OVERALL	254
7	TADJAE	OVERALL	254
8	TDOSE	OVERALL	254
9	TDURD	OVERALL	254
10	TPDOSE	OVERALL	254

```
# Count rows per by group using dplyr, producing the same result as count(adex, PARAMCD, PARCAT1)
adex %>% group_by(PARAMCD, PARCAT1) %>% summarize(n = n(), .groups = "drop")
```

```
# A tibble: 10 x 3
  PARAMCD PARCAT1      n
  <chr>    <chr>    <int>
1 ADJ      INDIVIDUAL    591
2 ADJAE    INDIVIDUAL    591
3 DOSE     INDIVIDUAL    591
4 DURD     INDIVIDUAL    591
5 PLDOSE   INDIVIDUAL    591
6 TADJ     OVERALL       254
7 TADJAE   OVERALL       254
8 TDOSE    OVERALL       254
9 TDURD    OVERALL       254
10 TPDOSE  OVERALL       254
```

Dose intensity can be calculated using the function `derive_param_doseint()`. The planned dose and administered dose are passed into the function and a new record is created with the dose intensity calculation. Again, only those variables specified in the `by_vars` argument will be populated in this new record.

```
dim(adex) # 4225 40
```

```
[1] 4225 40
```

```
adex <- adex %>%
  derive_param_doseint(
    by_vars = exprs(STUDYID, USUBJID, !!!adsl_vars),
    set_values_to = exprs(PARAMCD = "TNDOSINT"),
    tadm_code = "TDOSE",
    tpadm_code = "TPDOSE") # dim(adex) 4479 40
```

The default calculation for dose intensity is: Administered Doses / Planned Doses \* 100. Please see the `derive_param_doseint()` documentation to see how planned doses of 0 or NA are handled.

## Assign PARAMCD, PARAMN, etc. from Reference tables

To assign parameter level values such as PARAM, PARAMN, PARCAT1, etc., a lookup can be created to join to the source data.

```
# param_lookup not found

# adex <- derive_vars_merged(
#   adex,
```

```
# dataset_add = param_lookup,
# by_vars = exprs(PARAMCD)
# )

#count(adex, PARAMCD, PARAM, PARAMN)
```

## Derive Categorization Variables (AVALCATx)

### AVALCAT1

{admiral} does not currently have a generic function to aid in assigning AVALCATx/ AVALCAxN values. Below is a simple example of how these values may be assigned using the `dplyr::mutate` function:

```
adex <- adex %>%
  mutate(
    AVALCAT1 = case_when(
      PARAMCD %in% c("TDURD") & AVAL < 30 ~ "< 30 days",
      PARAMCD %in% c("TDURD") & AVAL >= 30 & AVAL < 90 ~ ">= 30 and < 90 days",
      PARAMCD %in% c("TDURD") & AVAL >= 90 ~ ">=90 days",
      PARAMCD %in% c("TDOSE", "TPDOSE") & AVAL < 1000 ~ "< 1000 mg",
      PARAMCD %in% c("TDOSE", "TPDOSE") & AVAL >= 1000 ~ ">= 1000 mg",
      TRUE ~ NA_character_
    )
  )
adex %>% select(USUBJID,VISIT,PARCAT1,PARAMCD,AVAL,AVALCAT1) %>% filter(!is.na(AVALCAT1)) %>%
```

```
# A tibble: 6 x 6
  USUBJID VISIT PARCAT1 PARAMCD AVAL AVALCAT1
  <chr>    <chr> <chr>    <chr>   <dbl> <chr>
1 01-701-1015 <NA> OVERALL TDOSE      0 < 1000 mg
2 01-701-1023 <NA> OVERALL TDOSE      0 < 1000 mg
3 01-701-1028 <NA> OVERALL TDOSE  1188 >= 1000 mg
4 01-701-1033 <NA> OVERALL TDOSE   756 < 1000 mg
5 01-701-1034 <NA> OVERALL TDOSE 14067 >= 1000 mg
6 01-701-1047 <NA> OVERALL TDOSE      0 < 1000 mg
```

### Assign ASEQ

#### ASEQ

- Analysis Sequence

The {admiral} function `derive_var_obs_number()` can be used to derive ASEQ. An example call is:

```
# Not working as PARAMN is not found
#adex <- derive_var_obs_number(
#  adex,
#  new_var = ASEQ,
#  by_vars = exprs(STUDYID, USUBJID),
#  order = exprs(PARCAT1, ASTDT, VISIT, VISITNUM, EXSEQ, PARAMN),
#  check_type = "error"
#)

#adex %>% select(USUBJID, ,VISIT,PARCAT1,PARAMCD,AVAL,ASTDT,ASEQ) %>% head()
```

If needed, the other ADSL variables can now be added:

```
# Deselect variables using negate_vars()
#adex <- adex %>%
#  derive_vars_merged(
#    dataset_add = select(adsl, !!!negate_vars(adsl_vars)),
#    by_vars = exprs(STUDYID, USUBJID)
#  )
```

## Add Labels and Attributes

Adding labels and attributes for SAS transport files is supported by the following packages:

- [metacore](#): establish a common foundation for the use of metadata within an R session.
- [metatools](#): enable the use of metacore objects. Metatools can be used to build datasets or enhance columns in existing datasets as well as checking datasets against the metadata.
- [xportr](#): functionality to associate all metadata information to a local R data frame, perform data set level validation checks and convert into a [transport v5 file\(xpt\)](#).

## References

[Creating a BDS Exposure ADaM](#)

[ADaM Subject-level Analysis - ADSL Dataset](#)