

```

*****
*****;
*
*      CLIENT: ModernaTX, Inc.
*      PROTOCOL: mRNA-1273-P301
*
*      PURPOSE: Create analysis dataset adslsf
*
*      INPUT FILES: SDTM domains
*      OUTPUT FILES: ADSLSF.sas7bdat
*
*      USAGE NOTES:
*
*****
*****;
* Copyright 2020 Pharmaceutical Product Development, Inc.
* All Rights Reserved.
*****
*****;

%include "madam.sas";

**Assign global macro variable DSETNAME to reflect the name of the final ADaM
dataset**;
%global DSETNAME;
%let dsetname = adslsf;

proc format;
    value $trtn
        'Placebo' = 1
        'mRNA-1273' = 2;

    value $reason
        'AE (specify)' = 'Adverse Event'
        'SAE (specify)' = 'Serious Adverse Event'
        'Death' = 'Death'
        'Lost To Follow-up' = 'Lost to Follow-Up'
        'Physician decision (specify)' = 'Physician Decision'
        'Pregnancy' = 'Pregnancy'
        'Protocol deviation (specify)' = 'Protocol Deviation'
        'Study Terminated By Sponsor' = 'Study Terminated by Sponsor'
        'Withdrawal of consent by participant (specify)' = 'Withdrawal of Consent
by Participant'
        'Due to SARS-COV-2' = 'Due to SARS-CoV-2'
        'Other' = 'Other';

    value rtpcr
        1 = 'Negative'
        2 = 'Positive';

run;

```

```

**Merge supplemental data onto parent domain**;
```

```

%revsupp(libin=trans,libout=work,ds=dm,supp=suppdm,outds=dm_all);
%revsupp(libin=trans,libout=work,ds=ds,supp=suppds,outds=ds_all);
%revsupp(libin=trans,libout=work,ds=rp,supp=supprp,outds=rp_all);

proc sort tagsort data=dm_all; by usubjid subjid; run;

/***** ds_all *****/

data ds_fail1_1;
    set ds_all;
    LENGTH DSTERM01 $200;
    _usubjid=usubjid;
    where (dsscat='SCREEN FAILURE' AND DSCAT="DISPOSITION EVENT");
    array _term (1) $200 DSTERM0;
    DSTERM01 = put(DSTERM0,$reason.);
    rename domain=_dom_fai;
    keep usubjid subjid dstermo dstermo1 dsstdtc domain dsterm: enrolllyn
dsdecod dscat dsscat;
run;
data ds_fail1_2;
    set ds_all;
    LENGTH DSTERM01 $200;
    _usubjid=usubjid;
    where (dscat="PROTOCOL MILESTONE" AND enrolllyn="N");
    array _term (1) $200 DSTERM0;
    DSTERM01 = put(DSTERM0,$reason.);
    rename domain=_dom_fai;
    keep usubjid subjid dstermo dstermo1 dsstdtc domain dsterm: enrolllyn
dsdecod dscat dsscat;
run;
proc sort data=ds_fail1_1 out=ds_fail1_11(keep=usubjid subjid) nodupkey;by usubjid
subjid;run;
proc sort data=ds_fail1_2;by usubjid subjid;run;
data ds_fail1_3;
    merge ds_fail1_11(in=a) ds_fail1_2(in=b);
    by usubjid subjid;
    if b and not a;
run;
data ds_fail1;
    set ds_fail1_1 ds_fail1_3;
run;

** Please let (b) (6) know if there is more than one SCFREAS.
    Note: If a subject screen failed twice and was not enrolled, the last
reason of screen failure will be included in the table. ;

proc sort data=adb.adsl out=adsl(keep=usubjid subjid);by usubjid subjid;run;
proc sort data=ds_fail1 out=ds_fail1;by usubjid subjid;run;

```

```

data ds_fail2;
    merge ds_fail1(in=a) adsl(in=b);
    by usubjid subjid;
    if a and not b;
run;

proc sort data=ds_fail2;by usubjid subjid dsstdtc dstermo dsterm;run;
data _ds_fail;
    set ds_fail2;
    by usubjid subjid dsstdtc;
    retain seq;
    if first.dsstdtc then seq=1;
    else seq = seq+1;
    if seq >1 then put "%str(alt)er_c: there is more than one SCFREAS";
    if last.dsstdtc;
run;

data _ds_fail;
    set _ds_fail;
    length scfreas scfreasp $200;
    if dscat= "DISPOSITION EVENT" and dsscat="SCREEN FAILURE" and
strip(dsdecod) ^= "OTHER" THEN scfreas = PROPCASE(dsterm);
    else if dscat= "DISPOSITION EVENT" and dsscat="SCREEN FAILURE" and dsdecod
="OTHER" THEN scfreas = "Other";

    if scfreas="Other" then scfreasp = strip(dsterm);
    keep usubjid subjid scfreas scfreasp;
run;

proc sort tagsort data=_ds_fail; by usubjid subjid; run;
/*****/

/***** rp_all *****/
data rp_all;
    set rp_all;
    if rptestcd='CHILDPOT' then _rstestcdn = 0;
    else if rptestcd='SURSTL' then _rstestcdn = 1;
    else if rptestcd='PM' then _rstestcdn = 2;
    else if rptestcd='PARTSTL' then _rstestcdn = 3;
    else if rptestcd='PREMENAR' then _rstestcdn = 4;
    else if rptestcd='OTHYSYS' then _rstestcdn = 5;
    else _rstestcdn = 9;
run;

proc sort tagsort data=rp_all;
    by usubjid _rstestcdn;
    where not missing(rpstresc) and _rstestcdn<9;

```

```

run;

data rp_chil rp_cbrsn;
    length CBRSN $200;
    set rp_all;
    by usubjid _rstestcdn;

    retain CBRSN;
    if first.usubjid and _rstestcdn=0 then CBRSN = '';
    else if first.usubjid then CBRSN = rptest;
    else if first._rstestcdn then CBRSN = catx('; ',CBRSN,rptest);

    if _rstestcdn=0 then output rp_chil;
    else if last.usubjid then output rp_cbrsn;
run;
/*****/

**INSERT CODE TO GENERATE DATASET**;
data all_merge_subjid1;
    merge dm_all(in=dm rename=(protver=_protver))
          _ds_fail(in=ds);
    by usubjid subjid;
    if ds AND DM;
run;

PROC SORT DATA=rp_chil NODUPKEY;BY USUBJID DOMAIN RPORRES;RUN;
data _all_merge(rename=(prevscr=_prevscr));
    merge all_merge_subjid1(in=a)
          rp_chil(rename=(domain=_dom_chil RPORRES=_RPORRES_chil)
keep=usubjid domain RPORRES)
          ;by usubjid;
    if a;
run;
PROC SORT DATA=rp_all NODUPKEY;BY USUBJID DOMAIN CBRSN;RUN;
data all_merge;
    merge _all_merge(in=a)
          rp_all(rename=(domain=_dom_cbrsn) keep=usubjid domain CBRSN)
          ;by usubjid;
    if a;
run;

data assign;* here.scrfailure;
    length RACESPY CBNRS $200 CHILDBP $10 PROTVER $25 PREVSCR $1;
    set all_merge;

    * Childbearing Potential;
    CHILDBP = _RPORRES_chil;
    CBNRS = CBRSN;

```

```

* Protocol Version;
  PROTVER = _protver;
* Was this participant screened previously;
  array _prevnum (1) $200 prevnum;
  PREVSCR = _PREVSCR;

  if RACE1='OTHER' then RACE1 = catx(' ', 'Other', RACEOTH);
  else RACE1 = propcase(RACE1);
  if RACE2='OTHER' then RACE2 = catx(' ', 'Other', RACEOTH);
  else RACE2 = propcase(RACE2);
  if RACE3='OTHER' then RACE3 = catx(' ', 'Other', RACEOTH);
  else RACE3 = propcase(RACE3);
  if RACE4='OTHER' then RACE4 = catx(' ', 'Other', RACEOTH);
  else RACE4 = propcase(RACE4);
  if RACE5='OTHER' then RACE5 = catx(' ', 'Other', RACEOTH);
  else RACE5 = propcase(RACE5);
  if RACE6='OTHER' then RACE6 = catx(' ', 'Other', RACEOTH);
  else RACE6 = propcase(RACE6);
  if UPCASE(RACE)="OTHER" then RACESPY = RACEOTH;
  ELSE if UPCASE(RACE)="MULTIPLE" THEN RACESPY = catx('
', RACE1, RACE2, RACE3, RACE4, RACE5, RACE6);
  RACESPY = tranwrd(RACESPY, ' Or ', ' or ');
  if index(RACESPY, "Native Hawaiian or Other Pacific Islander")>0
then RACESPY = tranwrd(RACESPY, 'Other', 'other');
run;

data adslsf;
  set assign;
  * Date of Informed Consent;
  %ISO2SAS(isodate=rficdtc, datec=_rficdtc, daten=RFICDT);
run;

```

****Generate final dataset by updating certain attributes. Optionally merge common variables & create sequence variable as needed**;**

```

%adam_dataset_update(
  ds=&DSETNAME,
  libin=work,
  libout=output,
  adsllib=adb,
  addcomvar=N,
  addseq=,
  dropinfmt=Y,
  mapspecfile=&ADAM_SPEC,
  maploc=&ADAM_SPEC_LOC,
  debug=N
);

```

```

**Generate Value Level Metadata values for the ad<xx> dataset**;
```

```

%* dod_vlm(
    type=ADAM,
    selmems=&DSETNAME,
    excmems=,
    specloc=&ADAM_SPEC_LOC,
    specname=&ADAM_SPEC,
    debug=N
);
```

```

**Generate Enhanced Controlled Terminology for the ad<xx> dataset**;
```

```

%* dod_enhcd_ct(
    type=ADAM,
    selmems=&DSETNAME,
    excmems=,
    specloc=&ADAM_SPEC_LOC,
    specname=&ADAM_SPEC,
    map_ct_nm=,
    map_ct_loc=,
    debug=N
);
```