

```

*****
*****;
** Program Name   : adae.sas                               *.
** Date Created  : 17Nov2021                               *.
** Programmer Name: (b) (4), (b)                          **.
** Purpose       : Create adae dataset                    **.
** Input data    : ae suppaex adsl                        **.
** Output data   : adae.sas7bdat                          **.
*****
*****;
options mprint mlogic symbolgen mprint symbolgen mlogic nocenter missing=" ";
**Setup the environment**;
%let
oprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_sdtm/saseng/cdisc3_0/data/
sdtm;
%let obla=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded/C4591001_BLA/saseng/cdisc3_0;/**Note:
This BLA data have been delivered during BLA Submission*/
%let
protori=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0;
%let
prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0/analy
sis/eSUB;

libname dataprot "&oprot." access=readonly;
libname dataeua "&obla./data" access=readonly;
libname datvprot "&protori./data_vai" access=readonly;
libname datvout "&prot./data_vai";

proc datasets lib=work kill;
run;

proc printto print="&prot./output/adae.rpt"
log="&prot./logs/adae.log" new;
run;

*****;
*Specification 1: Merging SDTM and Supplemental Datasets *;
*****;

data _spmdel_supp_dsin_subset_idvar1;
set dataprot.suppaex;
where idvar="AESEQ";

proc sort;
by studyid usubjid idvar idvarval;
run;

proc transpose data=_spmdel_supp_dsin_subset_idvar1
out=_spmdel_supp_dsin_idvar1_h;
by studyid usubjid idvar idvarval;
id qnam;
idlabel qlabel;
var qval;

```

```

quit;

data _ds2 (drop=idvar idvarval _NAME_ _LABEL_);
  set _spmdel_supp_dsin_idvar1_h;

  if idvar="AESEQ";
  AESEQ=input(idvarval, best12.);

proc sort;
  by STUDYID USUBJID AESEQ;
run;

proc sort data=dataprot.ae out=_ds1;
  by STUDYID USUBJID AESEQ;
run;

data _ae1;
  merge _ds1(in=d1) _ds2(in=d2);
  by STUDYID USUBJID AESEQ;
  if d1;
run;

*****;
* Specification 2 *;
* remove obs from REACTOGENICITY, obs with missing AETERM;
*;
* as all the AE do not have AESEV, drop AESEV;
*;
*****;

data _ae1;
  set _ae1;
  if aecat='REACTOGENICITY' then
    delete;
  AEDTC=' ';

  if ^missing(AETERM);
run;

*add variables DATCHGFL and DATAHGFC for change from EUA (EUA snapshot 25Mar2021 with the cutoff date
13Mar2021);

  data _spmdel_supp_dsin_subset_idvar1;
  set dataeua.suppae ; /*This is the SUPPAE dataset which were submmtted in the EUA Amendment submission
package for individuals 12 through 15 years of age in Apr-May2021, this dataset was not included in sBLA submission
package this time */
  where idvar="AESEQ";
  run;
proc sort data= _spmdel_supp_dsin_subset_idvar1;
  by studyid usubjid idvar idvarval;
quit;
proc transpose data= _spmdel_supp_dsin_subset_idvar1 out=_spmdel_supp_dsin_idvar1_h;
  by studyid usubjid idvar idvarval;
  id qnam;

```

```

idlabel qlabel;
var qval;
quit;

data _spmdel_temp(keep=AESEQ);
set dataeua.ae; /*This is the AE dataset which were submmtted in the EUA Amendment submission package for
individuals 12 through 15 years of age in Apr-May2021, this dataset was not included in sBLA submission package this
time */
run;
data _spmdel_suppds1 (drop=idvar idvarval _NAME_ _LABEL_);
set _spmdel_supp_ds1_idvar1_h;
if idvar="AESEQ";
AESEQ=input(idvarval,best12.);
run;
proc sort data=dataeua.ae out=_ds1;
by STUDYID USUBJID AESEQ;
run;
proc sort data=_spmdel_suppds1 out=_ds2;
by STUDYID USUBJID AESEQ;
run;
data _ae_bla;
merge _ds1(in=d1 ) _ds2(in=d2 ) ;
by STUDYID USUBJID AESEQ;
if d1;
run;

proc sql noprint undo _policy=none;
create table _ae1 as
select a.*,b.aespid as aespid1,b.AEDECOD as AEDECOD1,b.AEENDTC as AEENDTC1,b.AEOUT as
AEOUT1,b.AEENRTPT as AEENRTPT1,
b.AEACN as AEACN1,b.AECMGIV as AECMGIV1, b.AESUBJDC as AESUBJDC1, b.AENDGIV as
AENDGIV1,b.AECONTRT as AECONTRT1,
b.AERELTXT as AERELTXT1,b.AESTDTC as AESTDTC1,b.AETERM as AETERM1

from _ae1 as a left join _ae_bla as b
on a.usubjid=b.usubjid and a. aespid=b.aespid;
quit;

data _ae1;
set _ae1;
length DATCHGFL $1. DATACHGC $200.;
if missing(aespid1) then DATCHGFL="Y";
if DATCHGFL^="Y" then do;
if strip(aeodecod) ne strip(aeodecod1) then DATACHGC="AEDECOD";
if (strip(AEENDTC) ne strip(AEENDTC1) or strip(AEOUT ) ne strip(AEOUT1) or strip(AEENRTPT) ne
strip(AEENRTPT1) ) then do;
if DATACHGC ne "" then DATACHGC = trim(DATACHGC)||", AE outcome";
else DATACHGC = "AE outcome";
end;
if (strip(AEACN ) ne strip(AEACN1) or strip(AECMGIV) ne strip(AECMGIV1) or strip(AESUBJDC) ne
strip(AESUBJDC1) or strip(AENDGIV) ne strip(AENDGIV1) or strip(AECONTRT) ne strip(AECONTRT1) ) then
do;
if DATACHGC ne "" then DATACHGC = trim(DATACHGC)||", Actions to the AE";
else DATACHGC = "Actions to the AE";

```

```

end;
if (strip(AERELTXT) ne strip(AERELTXT1) ) then do;
  if DATACHGC ne "" then DATACHGC = trim(DATAACHGC)||", AERELTXT";
  else DATACHGC = "AERELTXT";
end;
if (strip(AESTDTC) ne strip(AESTDTC1) ) then do;
  if DATACHGC ne "" then DATACHGC = trim(DATAACHGC)||", AE onset date";
  else DATACHGC = "AE onset date";
end;
if (strip(AETERM) ne strip(AETERM1) and strip(aedecod) = strip(aedecod1)) then do;
  if DATACHGC ne "" then DATACHGC = trim(DATAACHGC)||", Minor AE term";
  else DATACHGC = "Minor AE term";
end;
if DATACHGC ne "" then DATAACHGC= trim(DATAACHGC)||" changed from EUA to sBLA";
end;
label DATCHGFL="New AE Reported post EUA snapshot Flag"
  DATAACHGC="AE data change post EUA snapshot";

run;
*****
* Specification 3 *;
* Handling of AE records with missing aedecod *;
*****

data g_a_dsin1a;
  attrib aedecod length=$200.;
  set _ae1 (where=((upcase(strip(AEDECOD))="UNCODED" or strip(AEDECOD)="")
    and ^missing(aeterm)));
  AEDECOD=upcase(strip(AETERM)) || "@@";

proc sort;
  by aedecod;
run;

data g_a_dsin1a;
  set g_a_dsin1a;
  by aedecod;
  retain aeptcd;

  If first.AEDECOD and AEPTCD=. then
    AEPTCD=_N_ + 9999999900;
  AEHLT="UNCODED TERM";
  AEHLGT="UNCODED TERM";
  AESOC="UNCODED TERM";
  AEBODSYS="UNCODED TERM";
run;

data _ae1;
  set g_a_dsin1a _ae1(where=(upcase(strip(AEDECOD)) ne "UNCODED" and
    strip(AEDECOD) ne "" and ^missing(aeterm)));
  aebodsys=upcase(aebodsys);
run;

*****

```

```

* Specification 4.1 *;
* Derive the analysis date and time variables;
*;
* Note: Time variables are not derived if time is not collected for any of the records *;
*****

```

```

data _ae2;
  length AESTDTC AEENDTC $20.;
  set _ae1;

  if ^missing(AESTDTC) then
    do;
      length yr $4 mm dd $2;
      yr=substr(AESTDTC, 1, 4);
      mm=substr(AESTDTC, 6, 2);
      dd=substr(AESTDTC, 9, 2);

      if yr ne '' then
        do;
          dflag=' ';

          if (dd eq " " or dd eq "-T") and mm ne " " then
            do;
              dd='01';
              dflag='D';
            end;

          if mm eq " " or mm eq "--" then
            do;
              mm='01';
              dd='01';
              dflag='M';
            end;
          newdate=(trim(left(yr))||'-'||trim(left(mm))||'-'||trim(left(dd)));
          ASTDT=input(newdate, ??is8601da.);
          format ASTDT date9.;
          ASTDTF=dflag;
        end;
      drop yr mm dd dflag newdate;
    end;

  if ^missing(AEENDTC) then
    do;
      length yr $4 mm dd $2;
      yr=substr(AEENDTC, 1, 4);
      mm=substr(AEENDTC, 6, 2);
      dd=substr(AEENDTC, 9, 2);

      if yr ne '' then
        do;
          dflag=' ';

          if (dd eq " " or dd eq "-T") and mm ne " " then
            do;

```

```

        fakedate=input(((trim(left(yr))||'-'||trim(left(mm))||'-'||'01')),
            ??is8601da.);
        format fakedate date9.;
        tempdate=intnx('month', fakedate, 1)-1;
        dd=strip(put(day(tempdate), best.));
        dflag='D';
    end;

    if (dd eq " " or dd eq "-T") and mm eq " " or mm eq "--" then
        do;
            mm='12';
            dd='31';
            dflag='M';
        end;
    newdate=(trim(left(yr))||'-'||trim(left(mm))||'-'||trim(left(dd)));
    AENDT=input(newdate, ??is8601da.);
    format AENDT date9.;
    AENDTF=dflag;
    drop fakedate tempdate;
end;
drop yr mm dd dflag newdate;
end;

if ^missing(AESTDTC) then
do;
    length hr mn sc $2 newtime $8;
    yr=substr(AESTDTC, 1, 4);
    hr=substr(AESTDTC, 12, 2);
    mn=substr(AESTDTC, 15, 2);
    sc=substr(AESTDTC, 18, 2);

    if yr ne ' ' then
        do;
            tflag=' ';

            if sc eq " " then
                do;
                    sc='00';
                    tflag='S';
                end;

            if mn eq " " then
                do;
                    mn='00';
                    tflag='M';
                end;

            if hr eq " " then
                do;
                    hr='00';
                    tflag='H';
                end;
            newtime=(trim(left(hr))||':'||trim(left(mn))||':'||trim(left(sc)));
            ASTTM=input(newtime, ??is8601tm.);

```

```
format ASTTM time8.;
ASTTMF=tflag;
drop tflag;
end;
drop yr hr mn sc newtime;
end;
```

```
if ^missing(AEENDTC) then
do;
```

```
length hr mn sc $2 newtime $8;
yr=substr(AEENDTC, 1, 4);
hr=substr(AEENDTC, 12, 2);
mn=substr(AEENDTC, 15, 2);
sc=substr(AEENDTC, 18, 2);
```

```
if yr ne ' ' then
do;
tflag=' ';
```

```
if sc eq " " then
do;
sc='59';
tflag='S';
end;
```

```
if mn eq " " then
do;
mn='59';
tflag='M';
end;
```

```
if hr eq " " then
do;
hr='23';
tflag='H';
end;
```

```
newtime=(trim(left(hr))||':'||trim(left(mn))||':'||trim(left(sc)));
AENTM=input(newtime, ??is8601tm.);
format AENTM time8.;
AENTMF=tflag;
drop tflag;
```

```
end;
drop yr hr mn sc newtime;
end;
```

```
run;
```

```
*****;
* Specification 4.2 *;
* Impute Missing Start/Stop Times *;
* Imputation of APxxSTM/APxxETM/APxxSDTM/APxxEDTM based on Time collected or missing. *;
* Merge AE SDTM and ADSL dataset (by USUBJID) *;
*****;
```

```
Data adsl;
```

```

Set datvprot.adsl;
If Missing(TR01STM) then
  _apx_TR01STM="00:00:00"t;
Else
  _apx_TR01STM=TR01STM;

If Missing(TR01ETM) then
  _apx_TR01ETM="23:59:59"t;
Else
  _apx_TR01ETM=TR01ETM;

If ^Missing(TR01SDT) then
  _apx_TR01SDTM=dhms(TR01SDT, 0, 0, _apx_TR01STM);

If ^Missing(TR01EDT) then
  _apx_TR01EDTM=dhms(TR01EDT, 0, 0, _apx_TR01ETM);
AP01SDT=datepart(_apx_TR01SDTM);
AP01STM=timepart(_apx_TR01SDTM);

If Missing(TR02STM) then
  _apx_TR02STM="00:00:00"t;
Else
  _apx_TR02STM=TR02STM;

If Missing(TR02ETM) then
  _apx_TR02ETM="23:59:59"t;
Else
  _apx_TR02ETM=TR02ETM;

If ^Missing(TR02SDT) then
  _apx_TR02SDTM=dhms(TR02SDT, 0, 0, _apx_TR02STM);

If ^Missing(TR02EDT) then
  _apx_TR02EDTM=dhms(TR02EDT, 0, 0, _apx_TR02ETM);
AP02SDT=datepart(_apx_TR02SDTM);
AP02STM=timepart(_apx_TR02SDTM);

if ^Missing(_apx_TR02SDTM-1) then
  do;
    _apx_TR01EDTM=min((_apx_TR02SDTM-1), (_apx_TR01EDTM+((365)*86400)));
  end;
else
  _apx_TR01EDTM=_apx_TR01EDTM+((365)*86400);
AP01EDT=datepart(_apx_TR01EDTM);
AP01ETM=Timepart(_apx_TR01EDTM);
AP01SDTM=dhms(AP01SDT, 0, 0, AP01STM);
AP01EDTM=dhms(AP01EDT, 0, 0, AP01ETM);
Attrib AP01SDT Label="Period 01 Start Date" AP01EDT Label="Period 01 End Date"
  AP01STM Label="Period 01 Start Time" AP01ETM Label="Period 01 End Time"
  AP01SDTM Label="Period 01 Start Datetime" AP01EDTM
  Label="Period 01 End Datetime";
Format AP01SDT AP01EDT date9. AP01STM AP01ETM time8. AP01SDTM AP01EDTM
  datetime20.;
_apx_TR02EDTM=_apx_TR02EDTM+((365)*86400);

```



```

AP02EDT=datepart(_apx_TR02EDTM);
AP02ETM=Timepart(_apx_TR02EDTM);
AP02SDTM=dhms(AP02SDT, 0, 0, AP02STM);
AP02EDTM=dhms(AP02EDT, 0, 0, AP02ETM);
Attrib AP02SDT Label="Period 02 Start Date" AP02EDT Label="Period 02 End Date"
  AP02STM Label="Period 02 Start Time" AP02ETM Label="Period 02 End Time"
  AP02SDTM Label="Period 02 Start Datetime" AP02EDTM
  Label="Period 02 End Datetime";
Format AP02SDT AP02EDT date9. AP02STM AP02ETM time8. AP02SDTM AP02EDTM
  datetime20.;
run;

proc sort data=_ae2 out=_ds1;
  by USUBJID;
run;

proc sort data=adsl out=_ds2
(keep=Usubjid SUBJID SITEID ARM ARMCD ACTARM ACTARMCD AGE AGEU AGEGR1 AGEGR1N
  AGEGR2 AGEGR2N AGEGR3 AGEGR3N AGEGR4 AGEGR4N RACE RACEN SEX SEXN ETHNIC
  ETHNICN COUNTRY ARACE ARACEN TRTSDT TRTSTM TRTSDTM TRTEDT TRTETM TRTEDTM
  TRT01A TRT01AN TRT01P TRT01PN TRT01A TRT01AN TRT02A TRT02AN TRT01P TRT01PN
  TRT02P TRT02PN TR01SDT TR01STM TR01SDTM TR01EDT TR01ETM TR01EDTM TR02SDT
  TR02STM TR02SDTM TR02EDT TR02ETM TR02EDTM VAX101DT VAX101TM VAX102DT VAX102TM
  SAFFL COHORT COHORTN DOSALVL DOSALVLN DOSPLVL DOSPLVLN VAX101 VAX102 V01DT
  V02DT RANDFL PHASE PHASEN DS30KFL COVBLST /*MULENRFL*/
  UNBLNDDT /*PROCGR1 PROCGR1N*/
  VAX10U VAX201 VAX202 VAX10UDT VAX10UTM VAX201DT VAX202DT VAX201TM VAX202TM
  V03DT V04DT TRT02A TRT02AN TRT02P TRT02PN HIVFL DS3KFL EOTXDCDT BDCSRDT
  X1CSRDT FUP1UNB FPX1CUT FUNBCUT FUP1CUT AGETR01 TR01SDTM TR01EDTM TR02SDTM
  TR02EDTM VAX20U VAX20UDT AP01SDT AP01STM AP01SDTM AP01EDT AP01ETM AP01EDTM
  AP02SDT AP02STM
  AP02SDTM AP02EDT AP02ETM AP02EDTM TR01SDT TR01STM TR01SDTM TR01EDT TR01ETM
  TR01EDTM TR02SDT TR02STM TR02SDTM TR02EDT TR02ETM TR02EDTM V02OBDT RACEGR1
  RACEGR1N);
  by USUBJID;
run;

data _ae3;
  merge _ds1(in=d1) _ds2(in=d2);
  by USUBJID;
  if d1;
  if agegr4n=1;
run;

data _ae4;
  set _ae3;

  if (missing(AESTDTC) and missing(AEENDTC) and missing(AEDTC)) then
  do;

    if ^missing(TRTSTM) then
    do;
      ASTTM=TRTSTM;
      ASTTMF='H';

```

```

        end;
        ASTDT=TRTSDT;
        ASTDTF='Y';
    end;

if ^missing(AENDTF) then
    AENDT=.;

if AENTMF in ('M' 'H') then
    do;
        AENTM=.;
        AENTMF="";
    end;
vax103dt=vax10udt;
vax103tm=0;
vax104dt=vax201dt;
vax104tm=vax201tm;
vax105dt=vax202dt;
vax105tm=vax202tm;
run;

%macro anadt;
    data _ae5(drop=TmpAstdtf);
        set _ae4;
        array _dte(5);
        TmpAstdtf=Astdtf;

        %do i=1 %to 5;
            _dte[&i]=VAX1%sysfunc(putn(&i, Z2.))DT;

            if ^missing(ASTDT) and TmpASTDTF='D' then
                do;

                    if (month(ASTDT)=month(_dte[&i]) and year(ASTDT)=year(_dte[&i]))
                        and (missing(AENDT) or AENDT > _dte[&i]) then
                            do;
                                ASTDT=_dte[&i];
                                TmpAstdtf="X";
                            end;
                end;

            if ^missing(ASTDT) and TmpASTDTF='M' then
                do;

                    if year(ASTDT)=year(_dte[&i]) then
                        do;

                            if .<AENDT < _dte[&i] then
                                do;
                                    ASTDT=intnx('year', _dte[&i], 0);
                                end;
                            else if _dte[&i] le AENDT or missing(AENDT) then
                                do;
                                    ASTDT=_dte[&i];
                                end;
                            end;
                end;
        %end;
    run;
%end;

```

```

        end;
        TmpAstdtf="Y";
    end;
end;
%end;
run;

%mend anadt;

%anadt;

data _ae6;
    set _ae5;
    vax103dt=vax10udt;
    vax103tm=vax10utm;
    vax104dt=vax201dt;
    vax104tm=vax201tm;
    vax105dt=vax202dt;
    vax105tm=vax202tm;
    ASTDTM=dhms(ASTDT, 0, 0, ASTTM);
    format ASTDTM datetime20.;
    AENDTM=dhms(AENDT, 0, 0, AENTM);
    format AENDTM datetime20.;

    if astdt=vax101dt and asttmf='H' then
        astdtm=dhms(vax101dt, 0, 0, vax101tm);

    if astdt=vax102dt and asttmf='H' then
        astdtm=dhms(vax102dt, 0, 0, vax102tm);

    if astdt=vax103dt and asttmf='H' then
        astdtm=dhms(vax103dt, 0, 0, vax103tm);

    if astdt=vax104dt and asttmf='H' then
        astdtm=dhms(vax104dt, 0, 0, vax104tm);

    if astdt=vax105dt and asttmf='H' then
        astdtm=dhms(vax105dt, 0, 0, vax105tm);

    if (astdtm>=dhms(vax101dt, 0, 0, vax101tm)>. or (astdtm=. and
        astdt>=vax101dt>.) then
        do;
            ASTDY=astdt-vax101dt+(astdt>=vax101dt);
            AENDY=aendt-vax101dt+(aendt>=vax101dt);
        end;

    if (astdtm>=dhms(vax102dt, 0, 0, vax102tm)>. or (astdtm=. and
        astdt>=vax102dt>.) then
        do;
            ASTDY=astdt-vax102dt+(astdt>=vax102dt);
            AENDY=aendt-vax102dt+(aendt>=vax102dt);
        end;

    if (astdtm>=dhms(vax103dt, 0, 0, vax103tm)>. or (astdtm=. and

```

```

astdt>=vax103dt>.) then
  do;
    ASTDY=astdt-vax103dt+(astdt>=vax103dt);
    AENDY=aendt-vax103dt+(aendt>=vax103dt);
  end;

if (astdtm>=dhms(vax104dt, 0, 0, vax104tm)>. or (astdtm=. and
astdt>=vax104dt>.) then
  do;
    ASTDY=astdt-vax104dt+(astdt>=vax104dt);
    AENDY=aendt-vax104dt+(aendt>=vax104dt);
  end;

if (astdtm>=dhms(vax105dt, 0, 0, vax105tm)>. or (astdtm=. and
astdt>=vax105dt>.) then
  do;
    ASTDY=astdt-vax105dt+(astdt>=vax105dt);
    AENDY=aendt-vax105dt+(aendt>=vax105dt);
  end;

if ASTDY=. and astdt ne . then
  do;
    ASTDY=astdt-vax101dt+(astdt>=vax101dt);
    AENDY=aendt-vax101dt+(aendt>=vax101dt);
  end;
ADURN=(AENDTM - ASTDTM) / (24*60*60);
ADURU='DAYS';
ADURN=round(ADURN, 0.1);
_DAYS=intck('days', ASTDT, AENDT) + 1;
ADURN=_DAYS;
ADURU='DAYS';
ADURN=round(ADURN, 0.1);

if missing(ASTDT) then
  ASTDY=.;

if missing(AENDT) then
  AENDY=.;

if missing(ASTDT) or ^missing(ASTDTF) or missing(AENDT) then
  do;
    ADURN=.;
    ADURU="";
  end;
drop vax103dt vax104dt vax104tm vax105dt vax104tm;
run;

*****
* Specification 5 *;
* create formats *;
*****

proc format;
  invalue AEREL "RELATED"=1 "

```

```

        POSSIBLY RELATED"=2
        "NOT RELATED"=3 "UNLIKELY RELATED"=4;
invalue RELGR "RELATED"=1 "NOT RELATED"=2;
invalue AESEV "MILD"=1 "MODERATE"=2 "SEVERE"=3;
invalue AETOXGR "1"=1 "2"=2 "3"=3 "4"=4 "5"=5 "MISSING OR UNKNOWN"=99;
run;

*****;
* Specification 6 *;
* Derive phase variables and period variables *;
*****;

Data Adsl_Phase;
  Set adsl;
  Length _b_Aphase $20.;
  Array _a_trta[*] Trt01A Trt02A;
  Array _a_trtan[*] Trt01AN Trt02AN;
  Array _a_trtp[*] Trt01P Trt02P;
  Array _a_trtpn[*] Trt01PN Trt02PN;
  Array _a_ApSdt[*] Ap01Sdt Ap02Sdt;
  Array _a_ApEdt[*] Ap01Edt Ap02Edt;
  Array _a_ApStm[*] Ap01Stm Ap02Stm;
  Array _a_ApEtm[*] Ap01Etm Ap02Etm;
  Array _a_ApSdtm[*] Ap01Sdtm Ap02Sdtm;
  Array _a_ApEdtm[*] Ap01Edtm Ap02Edtm;
  _b_subj_prcnt=n(_a_trtan[1], _a_trtan[2]);

do i=1 to 2;
  _b_trta=_a_trta[i];
  _b_trtan=_a_trtan[i];
  _b_trtp=_a_trtp[i];
  _b_trtpn=_a_trtpn[i];
  _b_apsdt=_a_apsdt[i];
  _b_apedt=_a_apedt[i];
  format _b_apsdt _b_apedt date9.;
  _b_apstm=_a_apstm[i];
  _b_apetm=_a_apetm[i];
  _b_apsdtm=_a_apsdtm[i];
  _b_apedtm=_a_apedtm[i];
  format _b_apstm _b_apetm time8. _b_apsdtm _b_apedtm datetime20.;
  _b_aperiod=i;
  _b_APhase="TREATMENT "||strip(put(_b_aperiod, z2.));

  if ^missing(_a_trta[i]) then
    Output adsl_phase;

  if i=1 then
    do;
      _b_apsdt=BrthDt;
      _b_apstm=0;
      _b_apsdtm=dhms(BrthDt, 0, 0, 0);
      _b_apedtm=_a_apsdtm[i]-1;
      _b_apedt=datepart(_b_apedtm);
      _b_apetm=timepart(_b_apedtm);
    end;
end;

```

```

    _b_Aperiod=i;
    _b_Aphase="PRE-TREATMENT";
    Output adsl_phase;
end;

if i=_b_subj_prdct then
do;
    _b_apsdtm=_a_apedtm[i]+1;
    _b_apedtm=_a_apedtm[i]+9999*86400;
    _b_apsdt=datepart(_b_apsdtm);
    _b_apedt=datepart(_b_apedtm);
    _b_apstm=timepart(_b_apsdtm);
    _b_apetm=timepart(_b_apedtm);
    _b_Aperiod=i;
    _b_Aphase="FOLLOW-UP";
    Output adsl_phase;
end;
end;
run;

Proc Sort Data=adsl_phase(Keep=usubjid _b.);
  By Usubjid descending _b_apsdtm;
Run;

Data Adsl_Offdrug;
  Set Adsl_Phase;
  By Usubjid;
  Where substr(_b_Aphase, 1, 9)="TREATMENT";
  _lag_b_apsdtm=lag(_b_apsdtm);

  if first.usubjid then
    _lag_b_apsdtm=.;
  format _lag_b_apsdtm datetime20.;
Run;

Proc Sort Data=Adsl_Offdrug(Keep=usubjid _b_:_lag_);
  By Usubjid _b_apsdtm;
Run;

Data Adsl_Offdrug;
  Set Adsl_Offdrug;
  where substr(_b_Aphase, 1, 9)="TREATMENT";
  _b_Aphase=tranwrd(_b_Aphase, "TREATMENT", "OFFDRUG");

  if ^missing(_lag_b_apsdtm) then
    _b_diff_dt=_lag_b_apsdtm - _b_apedtm;

  If _b_diff_dt>1 then
do;
    _b_apsdtm=_b_apedtm+1;
    _b_apedtm=_lag_b_apsdtm-1;
    _b_apsdt=datepart(_b_apsdtm);
    _b_apedt=datepart(_b_apedtm);
    _b_apstm=timepart(_b_apsdtm);

```

```

        _b_apetm=timepart(_b_apedtm);
    output;
End;
Run;

Data Adsl_Phase;
    Set Adsl_Phase Adsl_Offdrug;

Proc Sort;
    By usubjid descending _b_Aperiod _b_Aphase _b_apsdtm;
Run;

Data Adsl_Phase
(rename=( _b_trtan=TRTAN _b_trta=TRTA _b_trtpn=TRTPN _b_trtp=TRTP
    _b_apsdt=APHASDT _b_apedt=APHAEDT _b_apstm=AphaStm _b_apetm=AphaEtm
    _b_apsdtm=AphaSdtm _b_apedtm=AphaEdtm _b_Aphase=Aphase _b_Aperiod=Aperiod
    _b_AperiodC=AperiodC) );
Set Adsl_Phase;
By Usubjid descending _b_Aperiod;
_lag_b_trtan=lag(_b_trtan);
_lag_b_trtpn=lag(_b_trtpn);
_lag_b_trta=lag(_b_trta);
_lag_b_trtp=lag(_b_trtp);
_lag_b_Aperiod=lag(_b_aperiod);

if first.usubjid then
do;
    _lag_b_trtan=.;
    _lag_b_trtpn=.;
    _lag_b_trta=.;
    _lag_b_trtp=.;
    _lag_b_Aperiod=.;
end;

if index(_b_Aphase, "OFFDRUG") then
do;
    _b_Trtan=_lag_b_trtan;
    _b_Trtpn=_lag_b_trtpn;
    _b_Trta=_lag_b_trta;
    _b_Trtp=_lag_b_trtp;
    _b_Aperiod=_lag_b_Aperiod;
end;
_b_Aperiodc="Period "||strip(put(_b_Aperiod, z2.));
run;

Proc Sort Data=Adsl_Phase(drop=.);
    By Usubjid Aperiod AphaSdt AphaStm;
Run;

Data Adsl_Treatment(keep=UsubjId Aperiod Aperiodc AperiS: AperiE.);
    Set Adsl_Phase;

If substr(APhase, 1, 9)="TREATMENT";
AperiSdt=AphaSdt;

```

```
AperEdt=AphaEdt;
Aperstm=AphaStm;
Aperetm=AphaEtm;
Apersdtm=AphaSdtm;
Aperedtm=AphaEdtm;
```

```
Proc Sort;
  By UsubjId Aperiod;
Run;
```

```
Data Adsl_Phase;
  Attrib APERIOD Label="Period" APERIODC Label="Period (C)" Length=$20. APHASE
    Label="Phase" TRTP Label="Planned Treatment" TRTA Label="Actual Treatment"
    TRTPN Label="Planned Treatment (N)" TRTAN Label="Actual Treatment (N)"
    APHASDT Label="Phase Start Date" Format=date9.
  APHAEDT Label="Phase End Date" Format=date9.
  APERSDT Label="Period Start Date" Format=date9.
  APEREDT Label="Period End Date" Format=date9.
  APHASDTM Label="Phase Start Date/Time" Format=datetime20.
  APHAEDTM Label="Phase End Date/Time" Format=datetime20.
  APERSDTM Label="Period Start Date/Time" Format=datetime20.
  APEREDTM Label="Period End Date/Time" Format=datetime20.
  APHASTM Label="Phase Start Time" Format=time8.
  APHAETM Label="Phase End Time" Format=time8.
  APERSTM Label="Period Start Time" Format=time8.
  APERETM Label="Period End Time" Format=time8.;
  Merge Adsl_Phase(in=a) Adsl_Treatment;
  By Usubjid Aperiod;
```

```
  If a;
Run;
```

```
Data adsl_flg;
  Retain USUBJID TRTPN TRTP TRTAN TRTA APERIOD APERIODC APHASE APHASDT APHAEDT
    APERSDT APEREDT APHASTM APHAETM APHASDTM APHAEDTM APERSTM APERETM APERSDTM
    APEREDTM;
  Set Adsl_Phase;
Run;
```

```
Proc DataSets Lib=Work;
  Delete Adsl_OffDrug Adsl_Treatment Adsl_Phase;
Quit;
```

```
data adsl_flg(rename=(usubjid=usubjid1));
  set adsl_flg;
run;
```

```
data _ae9;
  set _ae6;
```

```
if asttm ne . and asttmf ne 'H' then
  _ASTDTMV=dhms(ASTDT, 0, 0, ASTTM);
else
  _ASTDTMV=dhms(ASTDT, 23, 59, 59);
```



```

format _ASTDTMV datetime20.;
run;

proc sql;
  create table _ae14 as select a.* , b.* from _ae9 as a left join adsl_flg as b
    on a.usubjid=b.usubjid1 and (b.APHASDTM <=a._ASTDTMV <=b.APHAEDTM);
quit;

*****
* Specification 7 *;
* Derivation of Causality Group *;
*****

data _ae16 _ae15;
  length relgr1 RELGR2 RELGR3 RELGR4 $30.;
  set _ae14(drop=usubjid1);

  if AECAT ^= 'MEDICATION ERROR' then
    do;
      AERELN=input(aerel, ?? aerel.);
      arel=aerel;
      ARELN=input(arel, ?? aerel.);

      if upcase(arel) in ('RELATED', 'POSSIBLY RELATED') then
        RELGR1='RELATED';
      else if upcase(arel) in ('NOT RELATED', 'UNLIKELY RELATED') then
        RELGR1='NOT RELATED';
      RELGR1N=input(RELGR1, ?? RELGR.);

      If upcase(arel) in ('RELATED', 'POSSIBLY RELATED') then
        RELGR2='RELATED';
      Else If upcase(arel) in ('NOT RELATED', 'UNLIKELY RELATED') then
        RELGR2='NOT RELATED';

      If upcase(arel)='RELATED' then
        RELGR3='RELATED';
      Else If upcase(arel)='NOT RELATED' then
        RELGR3='NOT RELATED';

      If upcase(arel)='RELATED' then
        RELGR4='RELATED';
      Else If upcase(arel) in ('NOT RELATED', 'UNLIKELY RELATED',
        'POSSIBLY RELATED') then
        RELGR4='NOT RELATED';
      RELGR2N=input(RELGR2, ?? RELGR.);
      RELGR3N=input(RELGR3, ?? RELGR.);
      RELGR4N=input(RELGR4, ?? RELGR.);
      label RELGR1='Pooled Causality Group 1'
        RELGR1N='Pooled Causality Group 1 (N)' RELGR2='Pooled Causality Group 2'
        RELGR2N='Pooled Causality Group 2 (N)' RELGR3='Pooled Causality Group 3'
        RELGR3N='Pooled Causality Group 3 (N)' RELGR4='Pooled Causality Group 4'
        RELGR4N='Pooled Causality Group 4 (N)';
      output _ae16;
    end;
end;

```

```

else
  do;
    output _ae15;
  end;
run;

data emerge;
  set _ae16;
  aetoxgrn=input(strip(aetoxgr), ?? aetoxgr.);

  if AETOXGRN ^=. then
    ATOXGR="GRADE "|| strip(AETOXGR);
  aetoxgrn=aetoxgrn;

  if (ASTDTM ne . and ASTDTM < TRTSDTM) or (ASTDT < TRTSDT) then
    PREFL='Y';

proc sort;
  by USUBJID AETERM ASTDTM APHASDTM;
run;

*****
* Specification 8 *;
* Derive AETPDOS and AEIMMFL *;
*****

data AEDOSZ AEDOSNZ;
  length ex_usubjid $22. ex_aeterm $44.;
  set emerge;
  by USUBJID AETERM ASTDTM APHASDTM;
  retain EX_USUBJID ' ' EX_AETERM ' ' EX_ASTDTM 0;

  if USUBJID=EX_USUBJID and AETERM=EX_AETERM and ASTDTM=EX_ASTDTM then
    do;
      DOSEAEON=0;
      DOSAEONU='mL';
      output AEDOSZ;
    end;
  else if first.ASTDTM and substr(APHASE, 1, 9) ^= 'TREATMENT' then
    do;
      DOSEAEON=0;
      DOSAEONU='mL';
      EX_USUBJID=USUBJID;
      EX_AETERM=AETERM;
      EX_ASTDTM=ASTDTM;
      output AEDOSZ;
    end;
  else
    do;
      output AEDOSNZ;
    end;
run;

data _ex(keep=USUBJID _EXSTD TM _EXENDTM EXDOSE EXDOSU EXCAT);

```

```

set dataprot.ex;

if ^missing(exstdtc);
  _EXSTDTM=input(exstdtc, is8601dt.);
  _EXENDTM=input(exendtc, is8601dt.);
  format _EXSTDTM _EXENDTM datetime19.;
run;

proc sql;
  create table _AEDOSNZ as select a.usubjid, a.ASTDTM, a.aeseq, b._EXSTDTM,
    b.EXDOSE as DOSEAEON , b.EXDOSU as DOSAEONU, (a.ASTDTM - b._EXSTDTM) as
    _diffdate, min(a.ASTDTM - b._EXSTDTM) as _mindate from AEDOSNZ as a left join
    _ex as b on a.usubjid=b.usubjid and b._EXSTDTM <=a.ASTDTM and
    b.EXCAT="INVESTIGATIONAL PRODUCT" group by a.usubjid, a.ASTDTM;
quit;

data _AEDOSNZ(keep=USUBJID AESEQ ASTDTM DOSEAEON DOSAEONU _minflg _EXSTDTM);
  set _AEDOSNZ;

  if _mindate=_diffdate;
  _minflg=1;
run;

proc sort data=_AEDOSNZ nodupkey;
  by USUBJID AESEQ ASTDTM DOSEAEON DOSAEONU _EXSTDTM;
run;

proc sort data=AEDOSNZ;
  by USUBJID AESEQ ASTDTM;
run;

data AEDOSNZ;
  merge _AEDOSNZ AEDOSNZ(drop=DOSEAEON DOSAEONU);
  by USUBJID AESEQ ASTDTM;
run;

data emerge1;
  set AEDOSNZ AEDOSZ;

  if ASTDTM >=TRTSDTM and _minflg=1 and not (length(aestdct)<=10) then
    do;
      AETPDOS=round(((ASTDTM - _EXSTDTM)/3600)*60, 0.01);
    end;

  if ^missing(ASTDTM) and ASTDTM < TRTSDTM and not (length(aestdct)<=10) then
    do;
      AETPDOS=round(((ASTDTM- TRTSDTM)/3600)*60, 0.01);
    end;

  if ^missing(AETPDOS) then
    do;

      if 0<=AETPDOS <=30 and aestdct ne " then
        do;

```

```

        AEIMMFL='Y';
        label AEIMMFL="Vaccine Studies AE Immediate flag";
    end;
end;
label AETPDOS="Time Post Dose";

proc sort;
    by USUBJID AETERM AEACN ASTDTM AENDTM DESCENDING APHASDTM APHAEDTM;
run;

*****
* Specification 9 *;
* For final ADaM - ADAE *;
* 1. Including Medication Error Records *;
* 2. Protocol specific derivation : VPHASE(N)*;
* 3. Sequencing and sorting of variables AND Label of ADAE *;
*****

data emerge2;
    set emerge1;
    by USUBJID AETERM AEACN ASTDTM AENDTM;

    if ^(FIRST.AENDTM) and upcase(AEACN)='DRUG WITHDRAWN' then
        do;
            AEACN="";
            AESUBJDC="";
        end;
run;

data aemerge;
    set emerge2 _ae15;
    ADESFL='N';

    if (index (upcase(AEACN), 'DRUG WITHDRAWN') > 0 or AESUBJDC='Y') then
        ADESFL='Y';
    drop aendtf;

    if prefl='Y' and ASTDTF in ('D', 'M') then
        do;
            ASTDTF=' ';
            ASTDT=.;
            ASTDY=.;
            ASTDTM=.;
        end;

proc sort;
    by usubjid;
run;

data aemerge1;
    set aemerge;
    format VAX101DTM VAX102DTM VAX10uDTM VAX201DTM VAX202DTM datetime20.;
    VAX101DTM=dhms(VAX101DT, 0, 0, VAX101TM);
    VAX102DTM=dhms(VAX102DT, 0, 0, VAX102TM);

```

```
VAX10uDTM=dhms(VAX10uDT, 0, 0, VAX10uTM);
VAX201DTM=dhms(VAX201DT, 0, 0, VAX201TM);
VAX202DTM=dhms(VAX202DT, 0, 0, VAX202TM);
length VPHASE $100.;
**pre treatment flag;
```

```
if .<ASTDTM<TRTSDTM or (ASTDTM=. and ASTDT<TRTSDT) or TRTSDT=. then
  PREFL="Y";
```

```
if index(upcase(cohort), '60 DAY') then
  do;
    d1tm1=105;
    d1tm6=238;
  end;
else
  do;
    d1tm1=58;
    d1tm6=191;
  end;
d2tm1=35;
d2tm6=168;
```

```
if ((astdt<unblnddt or astdt<vax201dt) or (UNBLNDDT=. and vax201dt=.)
or (astdtm<vax201dtm)) then
  do;
```

```
  if (prefl='Y' and vax101dt ne .) or (astdt ne . and vax101dt=.) then
    VPHASE='Pre-Vaccination ';
  else if (.<VAX101DT<=ASTDT<=V01DT and coalesce(vax102dt, vax10udt)=.)
    or .<VAX101DT<=ASTDT<coalesce(vax102dt, vax10udt)
    or (.<VAX101DTM<=ASTDTM<coalesce(vax102dtm, vax10udtm) and ASTDTF=") then
    VPHASE='Vaccination 1';
  else if ((astdtm >=coalesce(vax102dtm, vax10udtm)>.)
    or (.<coalesce(vax102dt, vax10udt)<=astdt<=v01dt and astdtm=.) and
    astdt<=v01dt then
    vphase='Vaccination 2';
  else if V01DT<ASTDT<=V02DT then
    VPHASE='Follow Up 1';
  else if ASTDT>V02DT then
    VPHASE='Follow Up 2';
end;
```

```
if astdt>=unblnddt>. or astdt>=vax201dt>. then
  do;
```

```
  if (astdt>=unblnddt and .<astdt<vax201dt) or ((astdt>=unblnddt and
    vax201dt=.) or ((astdtm<vax201dtm and asttmf ne 'H')) then
    VPHASE='After unblinding and before Vaccination 3';
  else if (vax202dt=. and .<vax201dt<=astdt and astdt<=v03dt) or (vax202dt
    ne . and vax201dt<=astdt<vax202dt) or (vax202dtm ne . and
    vax201dtm<=astdtm<vax202dtm) then
    vphase='Vaccination 3';
  else if (vax202dt ne . and (vax202dt<=astdt<=v03dt)) or (vax202dtm
    ne . and (vax202dtm<=astdtm<=v03dt)) then
```

```

    vphase='Vaccination 4';
else if .<v03dt<astdt and astdt<=v04dt>. then
    VPHASE='Follow Up 3';
else if .<v04dt<astdt then
    VPHASE='Follow Up 4';
end;

if VPHASE='Pre-Vaccination' then
    VPHASEN=0;
else if VPHASE='Vaccination 1' then
    VPHASEN=1;
else if VPHASE='Vaccination 2' then
    VPHASEN=2;
else if VPHASE='Follow Up 1' then
    VPHASEN=3;
else if VPHASE='Follow Up 2' then
    VPHASEN=99;
else if VPHASE='After unblinding and before Vaccination 3' then
    VPHASEN=4;
else if VPHASE='Vaccination 3' then
    VPHASEN=5;
else if VPHASE='Vaccination 4' then
    VPHASEN=6;
else if VPHASE='Follow Up 3' then
    VPHASEN=7;
else if VPHASE='Follow Up 4' then
    VPHASEN=100;

if vphasen>=3 then
do;

    if vax202dt ne . and ((astdt>vax202dt) or astdtm>=vax202dtm
        or (astdt=vax202dt and asttmf='H')) then
        VAXNO=4;
    else if vax202dt eq . and vax20udt ne . and vax201dt ne . and
        vax20udt > vax201dt and (astdt>=vax20udt) then
        VAXNO=4;
    else if vax201dt ne . and ((astdt>vax201dt) or astdtm>=vax201dtm
        or (astdt=vax201dt and asttmf='H')) then
        VAXNO=3;
    else if vax102dt ne . and ((astdt >=vax102dt>. and astdtm=.) or
        astdtm>=vax102dtm>.) then
        VAXNO=2;
    else if vax10udt ne . and ((astdt >=vax10udt>. and astdtm=.) or
        astdtm>=vax10udtm>.) then
        VAXNO=2;
    else if vax101dt ne . and ((astdt> vax101dt) or astdtm>=vax101dtm
        or (astdt=vax101dt and asttmf='H')) then
        VAXNO=1;

end;
else if vphasen not in (0, 2.5) then
    VAXNO=vphasen;
else if vphasen=2.5 then
    VAXNO=3;

```

```

if PREFL='N' and vphasen=0 then
  do;
    vphasen=1;
    vphase='Vaccination 1';
    VAXNO=1;
  end;
  ***in TREATMENT period flag;

if VPHASEN not in (., 0, 4, 99, 100) then
  INWDFL="Y";
  **removing imputation of date/time;

if ASTTMF in ('M' 'H') then
  do;
    ASTTM=.;
    ASTDTM=.;
    ASTTMF="";
  end;

if PREFL="Y" and ASTDTF in ('M' 'D') then
  do;
    ASTDTF=' ';
    ASTDT=.;
    ASTDTM=.;
  end;

if astdtf='Y' then
  do;
    astdtf="";
    astdt=.;
    astdtm=.;
    vphase="";
    vphasen=.;
    vaxno=.;
    inwdf="";
    aperiod=.;
    aperiodc="";
    astdy=.;
  end;
run;

proc sql;
  create table adae_
  (VPHASE char(100) "Vaccine Phase" , VPHASEN num(8) "Vaccine Phase(N)" ,
  INWDFL char(1) "Within Reporting Window Flag" , PREFL char(1)
  "AE Occured Prior to Vaccination Flag" , VAXNO num(8)
  "AE Occured after Which Vaccination");
quit;

data final;
  set adae_ aemerge1;

proc sort;

```

by USUBJID AEDECOD ASTDTM APHASDTM AESPID AESEQ AETERM;
quit;

data adae(label='Adverse Events Analysis Dataset');

```
retain SUBJID SITEID ARM ARMCD ACTARM ACTARMCD AGE AGEU AGEGR1 AGEGR1N AGEGR2
AGEGR2N AGEGR3 AGEGR3N AGEGR4 AGEGR4N RACE RACEN SEX SEXN ETHNIC ETHNICN
COUNTRY ARACE ARACEN TRTSDT TRTSTM TRTSDTM TRTEDT TRTETM TRTEDTM TRT01A
TRT01AN TRT01P TRT01PN VAX101DT VAX101TM VAX102DT VAX102TM SAFFL COHORT
COHORTN DOSALVL DOSALVLN DOSPLVL DOSPLVLN VAX101 VAX102 V01DT V02DT RANDFL
PHASE PHASEN DS30KFL COVBLST /*MULENRFL*/
UNBLNDDT /*PROCGR1 PROCGR1N*/
VAX10U VAX201 VAX202 VAX10UDT /*VAX10UTM*/
VAX201DT VAX202DT VAX201TM VAX202TM V03DT V04DT TRT02A TRT02AN TRT02P TRT02PN
HIVFL DS3KFL EOTXDCDT BDCSRDT X1CSRDT FUP1UNB FPX1CUT FUNBCUT FUP1CUT AGETR01
TR01SDTM TR01EDTM TR02SDTM TR02EDTM VAX20U VAX20UDT STUDYID USUBJID SUBJID
SITEID AESEQ AECAT AESPID AETERM AEDECOD
AEBDSYCD AEBODSYS AELLT AELLTCD AEPTCD AEHLT AEHLTCD AEHLGT AEHLGTCD AESOC
AESOCCD AESTDTC AESTDY AEENDTC AEENDY AEENRTPT AEENTPT AETOXGR DICTVER
ADESFL
ASTDT ASTDTF ASTDY ASTTM ASTDTM AENDT AENDY AENTM AENDTM ADURN ADURU AESER
AESCONG AESDISAB AESDTH AESHOSP AESLIFE AESMIE AEMERES AEREL AERELNST
AERELTXT AEACN AECMGIV AENDGIV AEOUT AESUBJDC AEREFID AERELN AREL ARELN
AETOXGRN ATOXGR ATOXGRN AEMEFL AETPDOS AEIMMFL APERIOD APERIODC APERSDT
APERSTM APERSDTM APEREDT APERETM APEREDTM VPHASE VPHASEN INWDFL PREFL VAXNO
VPHASE VPHASEN V02OBDT RACEGR1 RACEGR1N LSTCGDTC DATCHGFL DATAHGCG;
```

set final

```
(keep=SUBJID SITEID ARM ARMCD ACTARM ACTARMCD AGE AGEU AGEGR1 AGEGR1N
AGEGR2 AGEGR2N AGEGR3 AGEGR3N AGEGR4 AGEGR4N RACE RACEN SEX SEXN ETHNIC
ETHNICN COUNTRY ARACE ARACEN TRTSDT TRTSTM TRTSDTM TRTEDT TRTETM TRTEDTM
TRT01A TRT01AN TRT01P TRT01PN VAX101DT VAX101TM VAX102DT VAX102TM SAFFL
COHORT COHORTN DOSALVL DOSALVLN DOSPLVL DOSPLVLN VAX101 VAX102 V01DT V02DT
RANDFL PHASE PHASEN DS30KFL COVBLST /*MULENRFL*/
UNBLNDDT /*PROCGR1 PROCGR1N*/
VAX10U VAX201 VAX202 VAX10UDT /*VAX10UTM*/
VAX201DT VAX202DT VAX201TM VAX202TM V03DT V04DT TRT02A TRT02AN TRT02P TRT02PN
HIVFL DS3KFL EOTXDCDT BDCSRDT X1CSRDT FUP1UNB FPX1CUT FUNBCUT FUP1CUT AGETR01
TR01SDTM TR01EDTM TR02SDTM TR02EDTM VAX20U VAX20UDT STUDYID USUBJID SUBJID
SITEID AESEQ AECAT AESPID AETERM AEDECOD
AEBDSYCD AEBODSYS AELLT AELLTCD AEPTCD AEHLT AEHLTCD AEHLGT AEHLGTCD AESOC
AESOCCD AESTDTC AESTDY AEENDTC AEENDY AEENRTPT AEENTPT AETOXGR DICTVER
ADESFL
ASTDT ASTDTF ASTDY ASTTM ASTDTM AENDT AENDY AENTM AENDTM ADURN ADURU AESER
AESCONG AESDISAB AESDTH AESHOSP AESLIFE AESMIE AEMERES AEREL AERELNST
AERELTXT AEACN AECMGIV AENDGIV AEOUT AESUBJDC AEREFID AERELN AREL ARELN
AETOXGRN ATOXGR ATOXGRN AEMEFL AETPDOS AEIMMFL APERIOD APERIODC APERSDT
APERSTM APERSDTM APEREDT APERETM APEREDTM VPHASE VPHASEN INWDFL PREFL VAXNO
VPHASE VPHASEN V02OBDT RACEGR1 RACEGR1N LSTCGDTC DATCHGFL DATAHGCG);
```

```
label ASTTM='Analysis Start Time' AENTM='Analysis End Time'
ASTDTM='Analysis Start Date/Time' AENDTM='Analysis End Date/Time'
ADURU='Analysis Duration Units' ADURN='Analysis Duration (N)'
ASTDTF='Analysis Start Date Imputation Flag'
ASTDY='Analysis Start Relative Day' AENDT='Analysis End Date'
AEENDTC='End Date/Time of Adverse Event' AENDY='Analysis End Relative Day'
AERELN='Causality (N)' AESTDTC='Start Date/Time of Adverse Event'
```

FDA-CBER-2022-5812-0071688


```
AETOXGRN='Standard Toxicity Grade (N)' ATOXGR='Analysis Toxicity Grade'  
ATOXGRN='Analysis Toxicity Grade (N)' AREL='Analysis Causality'  
ARELN='Analysis Causality (N)' ASTDT='Analysis Start Date'  
PREFL='Pre-treatment Flag' ADESFL="Discontinued due to AE";
```

```
run;
```

```
options compress=yes;
```

```
data datvout.adae (label='Adverse Events Analysis Dataset');
```

```
  set adae;
```

```
run;
```

```
proc printto;
```

```
run;
```