

```
*****,  
** Program Name : adcevd.sas **;  
** Date Created : 10Mar2021 **;  
** Programmer Name : (b) (4), (b) (6) **;  
** Purpose : Create adcevd dataset **;  
** Input data : ce face vs ex suppce suppfac suppvvs adsl **;  
** Output data : adcevd.sas7bdat **;  
*****,
```

```
options mprint mlogic symbolgen mprint symbolgen mlogic nocenter missing=" ";  
proc datasets library=WORK kill nolist nodetails;  
quit;
```

```
**Setup the environment**;
```

```
%let  
oprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/bla_euaext_esub_sdtm/saseng/cdisc3_0;  
%let prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/euaext_esub_adam/saseng/cdisc3_0;  
libname dataprot "&oprot./data" access=readonly;  
libname datvprot "&prot./data_vai";
```

```
proc printto print="&prot./analysis/esub/output/adcevd.rpt"  
log="&prot./analysis/esub/logs/adcevd.log" new;  
run;
```

```
*****,  
*Specification 1.1 Input source data FACE and SUPPFAC *;  
*****,
```

```
data facevd;  
set dataprot.face;  
where facat in ("REACTOGENICITY EVENT", "REACTOGENICITY", "REACTOGENICITY - UNPLANNED  
ASSESSMENT");  
run;
```

```
proc sql noprint;  
create table domseq as select distinct min(FASEQ) as _minseq, max(FASEQ) as _maxseq from facevd;  
quit;
```

```
data suppfacvd;  
set dataprot.suppfac;  
do i=1 to n;  
set domseq point=i nobs=n;  
output;  
end;  
run;
```

```
data suppfacvd(drop=_minseq _maxseq);  
set suppfacvd;  
where (strip(uppercase(idvar))="FASEQ" and _minseq<=input(idvarval,? best.)<=_maxseq) or  
(strip(uppercase(idvar))="FACAT" and  
idvarval in ("REACTOGENICITY", "REACTOGENICITY - UNPLANNED ASSESSMENT")) or  
(missing(idvar));  
run;
```

```

data _spmdel_supp_dsin_subset;
  set suppfacevd;
run;

data _spmdel_sdtm_ds;
  set facevd;
run;

data _spmdel_supp_dsin_subset_idvar1;
  set _spmdel_supp_dsin_subset;
  where idvar="FASEQ";
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=FASEQ);
  set _spmdel_sdtm_ds;
run;

data _spmdel_suppds1 (drop=idvar idvarval _NAME_ _LABEL_);
  set _spmdel_supp_dsin_idvar1_h;
  if idvar="FASEQ";
  FASEQ=input(idvarval, best12.);
run;

proc sort data=_spmdel_sdtm_ds out=_ds1;
  by STUDYID USUBJID FASEQ;
run;

proc sort data=_spmdel_suppds1 out=_ds2;
  by STUDYID USUBJID FASEQ;
run;

data _spmdel_sdtm_temp_out1;
  merge _ds1(in=d1) _ds2(in=d2);
  by STUDYID USUBJID FASEQ;
  if d1;
run;

data facevd;
  set _spmdel_sdtm_temp_out1;
run;

proc datasets library=work noprint;
  delete _spmdel;;
quit;

```

```
*****;  
*Specification 1.2 Input source data VS and SUPPVS, *;  
*****;
```

```
data vsvd;  
  set dataprot.vs;  
  where vscat in ("REACTOGENICITY", "REACTOGENICITY EVENT", "REACTOGENICITY - UNPLANNED  
TEMPERATURE", "UNPLANNED TEMPERATURE");  
run;
```

```
proc sql noprint;  
  create table domseq as select distinct min(VSSEQ) as _minseq, max(VSSEQ) as _maxseq from vsvd;  
quit;
```

```
data suppvsvd;  
  set dataprot.suppvs;  
  do i=1 to n;  
    set domseq point=i nobs=n;  
    output;  
  end;  
run;
```

```
data suppvsvd(drop=_minseq _maxseq);  
  set suppvsvd;  
  where (strip(uppercase(idvar))="VSSEQ" and _minseq<=input(idvarval, ? best.)<=_maxseq) or  
(strip(uppercase(idvar))="VSCAT" and  
  idvarval in ("REACTOGENICITY", "REACTOGENICITY - UNPLANNED TEMPERATURE")) or  
(missing(idvar));  
run;
```

```
data _spmdel_supp_dsin_subset;  
  set suppvsvd;  
run;
```

```
data _spmdel_sdtm_ds;  
  set vsvd;  
run;
```

```
data _spmdel_supp_dsin_subset_idvar1;  
  set _spmdel_supp_dsin_subset;  
  where idvar="VSSEQ";  
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=VSSEQ);
  set _spmdel_sdtm_ds;
run;

data _spmdel_suppds1 (drop=idvar idvarval _NAME__LABEL__);
  set _spmdel_supp_dsin_idvar1_h;
  if idvar="VSSEQ";
  VSSEQ=input(idvarval, best12.);
run;

proc sort data=_spmdel_sdtm_ds out=_ds1;
  by STUDYID USUBJID VSSEQ;
run;

proc sort data=_spmdel_suppds1 out=_ds2;
  by STUDYID USUBJID VSSEQ;
run;

data _spmdel_sdtm_temp_out1;
  merge _ds1(in=d1) _ds2(in=d2);
  by STUDYID USUBJID VSSEQ;
  if d1;
run;

data vsvd;
  set _spmdel_sdtm_temp_out1;
run;

proc datasets library=work noprint;
  delete _spmdel.;
quit;

*****;
*Specification 1.3 Input source data CE and SUPPCE.      *;
*****;

data cevd;
  set dataprot.ce;
  where cecat in ("REACTOGENICITY", "REACTOGENICITY EVENT");
run;

proc sql noprint;
  create table domseq as select distinct min(CESEQ) as _minseq, max(CESEQ) as _maxseq from cevd;
quit;

data suppcevd;
  set dataprot.suppce;
  do i=1 to n;
    set domseq point=i nobs=n;
    output;
  end;
run;

```

```
data suppcevd(drop=_minseq _maxseq);
  set suppcevd;
  where (strip(upcase(idvar))="CESEQ" and _minseq<=input(idvarval,? best.)<=_maxseq) or
(strip(upcase(idvar))="CECAT" and
  idvarval in ("REACTOGENICITY")) or (missing(idvar));
run;
```

```
data _spmdel_supp_dsin_subset;
  set suppcevd;
run;
```

```
data _spmdel_sdtm_ds;
  set cevd;
run;
```

```
data _spmdel_supp_dsin_subset_idvar1;
  set _spmdel_supp_dsin_subset;
  where idvar="CESEQ";
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=CESEQ);
  set _spmdel_sdtm_ds;
run;
```

```
data _spmdel_suppds1 (drop=idvar idvarval _NAME__ LABEL__);
  set _spmdel_supp_dsin_idvar1_h;
  if idvar="CESEQ";
  CESEQ=input(idvarval, best12.);
run;
```

```
proc sort data=_spmdel_sdtm_ds out=_ds1;
  by STUDYID USUBJID CESEQ;
run;
```

```
proc sort data=_spmdel_suppds1 out=_ds2;
  by STUDYID USUBJID CESEQ;
run;
```

```
data _spmdel_sdtm_temp_out1;
  merge _ds1(in=d1) _ds2(in=d2);
  by STUDYID USUBJID CESEQ;
  if d1;
run;
```

```

data cevd;
  set _spmdel_sdtm_temp_out1;
run;

proc datasets library=work noprint;
  delete _spmdel;
quit;

data ex;
  set dataprot.ex;
  EXDIR=";
run;

*****;
*Specification 2.1 Pick up informtaion of FEVER from VS dataset *;
*****;

proc sql;
  create table vs_fever as select STUDYID, DOMAIN, USUBJID, VSLNKGRP as FALNKGRP, VSLNKID as
FALNKID, VSCAT as FACAT, VSSTRESN, VSSTRESU,
  VSTPTREF as FATPTREF, VSTPT as FATPT, VSDTC as FADTC, VSDY as FADY, VSSTAT as
FASTAT,VISIT, VISITNUM, CLTYP, VSEVAL as FAEVAL
  from vsvd where index(upcase(vstestcd), "TEMP") and ((index(upcase(vscat), 'REACTOGENICITY') and vstpt ne
") or (visit eq " and vsdte ne "))
  order by USUBJID, VSTPTREF, VSTPT;
quit;

proc sql;
  create table vs_fever2 as select a.*, b.FATPTREF as _FATPTREF from (select * from vs_fever where FAEVAL eq
'INVESTIGATOR') as a
  left join (select distinct USUBJID, scan(FADTC, 1, 'T') as _FADTC, FATPTREF from facevd) as b
  on a.USUBJID=b.USUBJID and a.FADTC=b._FADTC;
quit;

data facevd;
  set facevd vs_fever(where=(FAEVAL eq 'STUDY SUBJECT') in=a)
  vs_fever2(where=(FAEVAL eq 'INVESTIGATOR' and _FATPTREF ne ") in=b);

  if b then do;
    FACAT="REACTOGENICITY - UNPLANNED ASSESSMENT";
    FATPTREF=strip(_FATPTREF);
    FALNKGRP=strip(FATPTREF)||"-FEVER";
    FALNKID=upcase(compbl(catx('-', FATPTREF, "FEVER")));
  end;

  if a or b then do;
    FASCAT="SYSTEMIC";
    FAOBJ="FEVER";
    FATESTCD="OCCUR";
    FAGRPID=upcase(compbl(catx('-', FATPTREF, "SYSTEMIC")));

    if vsstresu="C" then do;
      if .<vsstresn<38 then FASTRESC="N";
      else if vsstresn>=38 then FASTRESC="Y";
    end;
  end;

```

```
        else FASTRESC=" ";
        if vsstresn>42 then FASTRESC="N";
    end;
    else if FASTAT="NOT DONE" then FASTRESC=" ";
    else put "Note: incorrect VSSTRESU - " USUBJID VSSTRESU;
end;
```

```
    if index(faobj, "HOSPITAL")<=0 and fatestd ne "REL";
run;
```

```
data facevd;
    set facevd;
    FADIR=";
    _fatpt=input(substr(FATPT, 4), 8.);
run;
```

```
*****;
*Specification 2.2 Merge FACEVD with EX dataset *;
*****;
```

```
data facevd1 facevd2 facevd3;
    length FATPTREF $100.;
    set facevd;

    if upcase(FASCAT)='ADMINISTRATION SITE' then output facevd1;
    else if upcase(FASCAT) in ('SYSTEMIC', 'MEDICATIONS GIVEN') then output facevd2;
    else output facevd3;
run;
```

```
proc sort data=facevd1;
    by usubjid fatptref;
quit;
```

```
data _ex(rename=(exlnkgrp=fatptref));
    set ex;
run;
```

```
proc sort data=_ex;
    by usubjid fatptref;
quit;
```

```
data facevd111;
    merge facevd1(in=a) _ex(keep=usubjid exdose extrt exdosu exstdtc exendtc fatptref);
    by usubjid fatptref;
    if a;
run;
```

```
proc sort data=facevd2;
    by usubjid fatptref;
quit;
```

```
data facevd2;
    merge facevd2(in=a) _ex(keep=usubjid exdose extrt exdosu exstdtc exendtc fatptref);
```

FDA-CBER-2022-5812-0072068

```

    by usubjid fatptref;
    if a;
run;

data facevd;
    set facevd111 facevd2 facevd3;
    FALNKGRP=upcase(FALNKGRP);

    if fatpt eq " then do;
        _fatpt=input(scan(fadtc, 1, 'T'), yymmdd10.)-input(scan(exstdtc, 1, 'T'), yymmdd10.)+1;
        fatpt="DAY "||strip(_fatpt);
    end;

    if _fatpt>7 and FAEVAL='INVESTIGATOR' and faobj ne "MEDICATIONS" then delete;
run;

**Apply cutoff to remove ed diary data after unblinding date**;
proc sort data=facevd;
    by usubjid;
run;

data UNBLNDDT;
    set datvprot.adsl;
    where UNBLNDDT ne .;
    keep usubjid UNBLNDDT;
run;

proc sort;
    by usubjid;
run;

data facevd;
    merge facevd(in=a) UNBLNDDT(in=b);
    by usubjid;
    if a and not(input(scan(fadtc, 1, "T"), ??yymmdd10.)>=UNBLNDDT>.);
    drop UNBLNDDT;
run;

*****
*Specification 2.3 Merge CEVD with EX dataset          *;
*****

proc sort data=cevd nodupkey;
    where cetptref ne ";
    by usubjid ceterm cetptref ceoccur cestdte ceendte;
run;

data cevd;
    set cevd;
    length CEDIR $50;
    CEDIR=";
run;

```

```

data cevd0(rename=(ceseq=srcseq domain=srcdom));
  set cevd;
  _celnkgrp=compbl(catx('-', CETPTREF, CELOC, CELAT));
  _celnkid=CETPTREF;
run;

data cevd1 cevd2 cevd3;
  set cevd0;

  if upcase(CESCAT)='ADMINISTRATION SITE' then output cevd1;
  else if (CESCAT)='SYSTEMIC' then output cevd2;
  else output cevd3;
run;

proc sort data=cevd1;
  by usubjid _CELNKID;
quit;

proc sort data=cevd2;
  by usubjid _CELNKID;
quit;

data ex1(rename=(exlnkid=_CELNKGRP));
  set ex;
run;

data ex2(rename=(exlnkgrp=_CELNKID));
  set ex;
run;

proc sort data=ex1;
  by usubjid _CELNKGRP;
quit;

proc sort data=ex2;
  by usubjid _CELNKID;
quit;

data cevd1;
  merge cevd1(in=a) ex2(keep=usubjid exdose extrt exdosu exstdtc exendtc _CELNKID exlnkid exloc exlat);
  by usubjid _CELNKID;
  if a;
run;

data celnkgrp;
  set cevd1;
  if exloc ne " and (celoc ne exloc or celat ne exlat);
run;

data cevd2;
  merge cevd2(in=a) ex2(keep=usubjid exdose extrt exdosu exstdtc exendtc _CELNKID);
  by usubjid _CELNKID;
  if a;

```

```

run;

data cevd0;
  set cevd1 cevd2 cevd3;
run;

*****;
* Specification 3 Duration Derivation ASTDT, AENDT, ADURN          *;
* First to Last Day where ASEV="          *;
* First to Last Day (for >21 Calipers Redness or Swelling ): ASEV='Severe'*;
*****;

proc sql;
  create table _temp1 as select distinct USUBJID, FAOBJ, FATPTREF, EXTRT, max(_fatpt) as last_fatpt
    from facevd where FASTAT="" and upcase(FAOBJ) not in ('REDNESS', 'SWELLING')
    group by USUBJID, FAOBJ, FATPTREF, EXTRT;

  create table _temp2 as select distinct USUBJID, FAOBJ, FATPTREF, EXTRT, max(_fatpt) as last_fatpt
    from facevd where FASTAT="" and upcase(FAOBJ) in ('REDNESS', 'SWELLING')
    group by USUBJID, FAOBJ, FATPTREF, EXTRT;
quit;

data _temp;
  set _temp1 _temp2;
run;

proc sql;
  create table facevd as select a.*, b.last_fatpt from facevd a left join _temp b
    on a.USUBJID=b.USUBJID and a.FAOBJ=b.FAOBJ and a.FATPTREF=b.FATPTREF and a.EXTRT=b.EXTRT;
quit;

data _ds1;
  set facevd;
  if ^missing(fadtc) and FASTRESC='Y' and FATESTCD='OCCUR' and index(upcase(faobj), 'MEDICATION')=0 and
  index (upcase(faobj), 'SWELLING')=0 and index (upcase(faobj), 'REDNESS')=0;
run;

data _ds2;
  set facevd;

  if ^missing(fadtc) and FASTRESN >=2.5 and index(FATESTCD,'DIAM')
  and (index (upcase(faobj), 'SWELLING')>0 or index (upcase(faobj), 'REDNESS')>0);
run;

data _ds3_21;
  set facevd;

  if ^missing(fadtc) and FASTRESN >=10.5 and index(FATESTCD,'DIAM')
  and (index (upcase(faobj), 'SWELLING')>0 or index (upcase(faobj), 'REDNESS')>0);
run;

data _ds3;
  set _ds1 _ds2;
run;

```

```

proc sort data=_ds3 out=_ds4;
  by USUBJID FATPTREF EXTRT FAOBJ _fatpt fady;
run;

proc sort data=_ds3_21 out=_ds4_21;
  by USUBJID FATPTREF EXTRT FAOBJ _fatpt fady;
run;

* Derive ASTDT, AENDT*;
data _ds4(drop=fa:);
  length AOCCURFL $1. ATPT $30. _CENDY 8.;
  set _ds4;
  by USUBJID FATPTREF EXTRT FAOBJ _fatpt fady;
  retain astdt ASTTM;

  if first.faobj then do;
    ASTDT=input(fadtc, ??is8601da.);
    format ASTDT date9.;
    ASTTM=.;

    if length(strip(fadtc))>12 then do;
      ASTTM=input(substr(fadtc, 12), ??is8601tm.);
    end;
    format ASTTM time8.;

    ASTDTM=DHMS(ASTDt, 0, 0, ASTTM);
    format ASTDTM datetime20.;
    label ASTDTM='Analysis Start Date/Time';
  end;

  if last.faobj then do;
    AOCCURFL='Y';
    CETPTREF=FATPTREF;
    CETERM=FAOBJ;

    if _fatpt < last_fatpt then do;
      _AENDT=input(substr(fadtc, 1, 10), is8601da.);
      format _aendt date9.;
    end;
    else if _fatpt=last_fatpt then do;
      _AENDT=.;
    end;
    _CESCAT=FASCAT;
    _CELNKGPR=FALNKGPR;
    ATPT=FATPT;
    _CENDY=FADY;
    output;
  end;
run;

proc sort data=_ds4;
  by usubjid CETPTREF CETERM extrt;
run;

```

* Derive ASTDT, AENDT for >21 Calipers Redness or Swelling*;

```
data _ds4_21(drop=fa:);
  length ATPT $30. _CENDY 8.;
  set _ds4_21;
  by USUBJID FATPTREF EXTRT FAOBJ _fatpt fady;
  retain astdt ASTTM;

  if first.faobj then do;
    ASTDT=input(fadtc, ??is8601da.);
    format ASTDT date9.;
    ASTTM=.;

    if length(strip(fadtc))>12 then do;
      ASTTM=input(substr(fadtc, 12), ??is8601tm.);
      end;
    format ASTTM time8.;

    ASTDTM=DHMS(ASTD, 0, 0, ASTTM);
    format ASTDTM datetime20.;
    label ASTDTM='Analysis Start Date/Time';
  end;

  if last.faobj then do;
    AOCCURFL='Y';
    CETPTREF=FATPTREF;
    CETERM=FAOBJ;
    ASEV="Severe";

    if _fatpt < last_fatpt then do;
      _AENDT=input(substr(fadtc, 1, 10), is8601da.);
      format _aendt date9.;
      end;
    else if _fatpt=last_fatpt then do;
      _AENDT=.;
      end;
    _CESCAT=FASCAT;
    _CELNKGPR=FALNKGPR;
    ATPT=FATPT;
    _CENDY=FADY;
    output;
  end;
run;

proc sort data=_ds4_21;
  by usubjid CETPTREF CETERM extrt;
run;

proc sort data=cevd0 out=cevd0;
  by usubjid CETPTREF CETERM extrt;
  where CESTAT="";
run;
```

* Derive ADURN First to Last Day*;

data cevd1;

```
merge cevd0(in=a) _ds4(in=b);  
by usubjid CETPTREF CETERM extrt;
```

```
if _AENDT eq . and (a and b) then do;
```

```
if upcase(ONGNXVIS)='Y' and missing(RCENDTC) then ADURN=.;  
else do;
```

```
if length(RCENDTC)=10 then AENDT=input(substr(RCENDTC,1,10),is8601da.);  
format aendt date9.;  
ADURN=max(AENDT, _AENDT) - ASTDT + 1;
```

```
if ADURN ne . then ADURU='DAYS';  
AENDT=max(AENDT, _AENDT);  
AENDY=max(CEENDY, _CENDY);
```

```
end;
```

```
end;
```

```
else if _AENDT ^=. or (b and not a) then do;
```

```
ADURN=_AENDT - ASTDT +1;  
ADURU='DAYS';  
AENDT=_AENDT;  
AENDY=_CENDY;  
CECAT=coalescec(CECAT, "REACTOGENICITY");  
CESCAT=coalescec(CESCAT, _CESCAT);  
CEPRESP=coalescec(CEPRESP, 'Y');  
CEOCCUR=coalescec(CEOCCUR, 'Y');  
CELNKGRP=coalescec(CELNKGRP, _CELNKGRP);
```

```
end;
```

```
format aendt date9.;
```

```
run;
```

* Derive ADURN First to Last Day (for >21 Calipers Redness or Swelling)*;

data cevd2;

```
merge cevd0(in=a) _ds4_21(in=b);  
by usubjid CETPTREF CETERM extrt;  
CECAT=coalescec(CECAT, "REACTOGENICITY");  
CESCAT=coalescec(CESCAT, _CESCAT);  
CEPRESP=coalescec(CEPRESP, 'Y');  
CEOCCUR=coalescec(CEOCCUR, 'Y');  
CELNKGRP=coalescec(CELNKGRP, _CELNKGRP);
```

```
if _AENDT= . and (a and b) then do;
```

```
if upcase(ONGNXVIS)='Y' and missing(RCENDTC) then ADURN=.;  
else do;
```

```
if length(RCENDTC)=10 then AENDT=input(substr(RCENDTC,1,10),is8601da.);  
format aendt date9.;  
ADURN=max(AENDT, _AENDT) - ASTDT + 1;
```

```
if ADURN ne . then ADURU='DAYS';  
AENDT=max(AENDT, _AENDT);  
AENDY=max(CEENDY, _CENDY);
```

FDA-CBER-2022-5812-0072074

```

        end;
    output;
end;
else if _AENDT ^= . or (b and not a) then do;
    ADURN=_AENDT - ASTDT +1;
    ADURU='DAYS';
    AENDT=max(AENDT, _AENDT);
    AENDY=_CENDY;
    output;
end;
format aendt date9.;
run;

proc sql;
    create table _ds5 as select USUBJID, CETPTREF, EXTRT, ADURN, CETERM , AENDT , AENDY, _cendy from
cevd1;

    create table _ds6 as select a.CETPTREF, a.adurn, a.ceterm, a.aendt, a.aendy, a._cendy, b.* from _ds5 as a left join
_ds3 as b
    on a.USUBJID=b.USUBJID and upcase(a.CETERM)=upcase(b.FAOBJ) and
upcase(a.CETPTREF)=upcase(b.FATPTREF) and a.EXTRT=b.EXTRT;
quit;

data _ds6;
    set _ds6;
    length AENDY _dur 8;
    _dur=1;

    if _fatpt=last_fatpt and AENDY > _CENDY then do;
        _dur=AENDY - _CENDY + 1;
    end;
run;

proc sql;
    create table _ds7 as select distinct USUBJID, CETPTREF, EXTRT, CETERM, AENDY, sum(_dur) as ADURTN
from _ds6
    group by USUBJID, CETPTREF, EXTRT, CETERM;

    create table cevd1 as select a.*, b.AENDY , ADURTN , 'DAYS' as ADURTU from cevd1 as a left join _ds7 as b
    on a.USUBJID=b.USUBJID and upcase(a.CETPTREF)=(b.CETPTREF) and a.EXTRT=b.EXTRT and
upcase(a.CETERM)=upcase(b.CETERM);
quit;

proc sql;
    create table _ds5_21 as select USUBJID, CETPTREF, EXTRT, ADURN, CETERM , AENDT , AENDY, _cendy
from cevd2;
    create table _ds6_21 as select * from _ds5_21 as a left join _ds3_21 as b
    on a.USUBJID=b.USUBJID and upcase(a.CETERM)=upcase(b.FAOBJ) and
upcase(a.CETPTREF)=upcase(b.FATPTREF) and a.EXTRT=b.EXTRT;
quit;

data _ds6_21;
    set _ds6_21;
    length AENDY _dur 8;

```

```

_dur=1;

if _fatpt=last_fatpt and AENDY > _CENDY then do;
    _dur=AENDY - _CENDY + 1;
end;

run;

proc sql;
    create table _ds7_21 as select distinct USUBJID, CETPTREF, EXTRT, CETERM, AENDY, sum(_dur) as ADURTN
    from _ds6_21
        group by USUBJID, CETPTREF, EXTRT, CETERM;
    create table cevd2 as select a.*, b.AENDY , ADURTN , 'DAYS' as ADURTU from cevd2 as a left join _ds7_21 as b
        on a.USUBJID=b.USUBJID and upcase(a.CETPTREF)=(b.CETPTREF) and a.EXTRT=b.EXTRT and
        upcase(a.CETERM)=upcase(b.CETERM);
quit;

data cevd1;
    set cevd1 cevd2;

run;

*****
* Specification 4 Derive analysis flags: KNOWVFL, EVENTFL  *;
* KNOWVFL-Known value flag                               *;
* EVENTFL-Event value flag                               *;
*****

proc sql;
    create table _knowfl as select distinct STUDYID, USUBJID, FALNKGRP, FATPTREF,FAOBJ, FASCAT, EXTRT,
    DOMAIN from facevd
        where upcase(fatestcd)='OCCUR' and upcase(fastresc) in ('Y', 'N') and index(upcase(facat), 'REACTOGENICITY')
        and (upcase(CLTYP)='DIARY CARD' or ^(fatestcd in ("MEDTFVPN" "REL") and faeval="INVESTIGATOR"))
        order by USUBJID, FATPTREF, FAOBJ, EXTRT;
quit;
proc sort data=_knowfl nodupkey;
    by USUBJID FATPTREF FAOBJ EXTRT;
run;

data _knowfl;
    set _knowfl;
    CEPRESP='Y';
    CETPTREF=FATPTREF;
    CETERM=FAOBJ;
    knowvfl='Y';

run;

proc sort data=cevd1;
    by usubjid CETPTREF CETERM;
quit;

proc sort data=_knowfl;
    by usubjid CETPTREF CETERM;
quit;

data cevd1;

```

```

merge cevd1(in=x) _knowfl(in=y);
by usubjid CETPTREF CETERM;
if x or y;
if x and not y then knowvfl='N';
if not x and y then do;
    CECAT="REACTOGENICITY";
    CESCAT=FASCAT;
    CELNKGRP=FALNKGRP;
end;
run;

proc sql;
    create table _EVENTFL as select distinct USUBJID, FATPTREF, FAOBJ, COUNT(fastresc) AS CNT1 from facevd
    where (fatestcd='OCCUR' and fastresc='Y' and upcase(faobj) not in ('REDNESS', 'SWELLING')) or (FASTRESN
>=2.5 and
    index(upcase(FATESTCD), 'DIAM')) and index(upcase(facat), 'REACTOGENICITY') and
(upcase(CLTYP)='DIARY CARD' or
    ^(fatestcd in ("MEDTFVPN" "REL") and faeval="INVESTIGATOR")) group by USUBJID, FATPTREF, FAOBJ;
quit;

proc sort data=cevd1;
    by usubjid CETPTREF CETERM;
quit;

proc sort data=_EVENTFL(rename=(FATPTREF=CETPTREF FAOBJ=CETERM));
    by usubjid CETPTREF CETERM;
quit;

data cevdf;
    merge cevd1(in=x) _eventfl;
    by usubjid CETPTREF CETERM;
    if x;
    EVENTFL='N';

    if cnt1 > 0 then EVENTFL='Y';
run;

data adcevd;
    set cevdf;
run;

*****;
* Specification 5 *;
* Merge with adsl dataset and keep necessary variables. *;
* Generate TRTx(N) . *;
* Impute stop date of meds with dose 2 date *;
*****;

%let g_adsl_vars=%str(STUDYID USUBJID SUBJID SITEID AGE AGEU SEX SEXN RACE RACEN ARACE
ARACEN SAFFL ARM ARMCD ACTARM ACTARMCD
TRTSDT TRTSTM TRTSDTM TRTEDT TRTETM TRTEDTM TRT01A TRT01AN TRT02A TRT02AN TRT01P
TRT01PN TRT02P TRT02PN TR01SDT TR01STM TR01SDTM TR01EDT TR01ETM TR01EDTM TR02SDT
TR02STM TR02SDTM TR02EDT TR02ETM TR02EDTM
VAX101DT VAX102DT COHORT COHORTN PHASE PHASEN DOSPLVL DOSPLVLN DOSALVL DOSALVLN

```

FDA-CBER-2022-5812-0072077

```
AGEGR1 AGEGR1N AGEGR2 AGEGR2N AGEGR4 AGEGR4N VAX101 VAX102
DS30KFL COVBLST MULENRFL HIVFL PEDREAFL UNBLNDDT DS3KFL REACTOFL AGETR01 VAX10U
VAX10UDT RANDFL RAND1FL SAF1FL SAF2FL);
```

```
Data work.adsl;
  Set datvprot.adsl;
run;
```

```
proc sort data=adcevd out=_ds1;
  by USUBJID;
run;
```

```
proc sort data=work.adsl out=_ds2;
  by USUBJID;
run;
```

```
data adcevd;
  merge _ds1(in=d1) _ds2(in=d2 keep=&g_adsl_vars.);
  by USUBJID;
  if d1;
run;
```

```
proc datasets library=work noprint;
  delete Allvars;
quit;
```

```
data adcevd;
  set adcevd;
```

```
  if CETPTREF="VACCINATION 1" and VAX102DT ne . and ASTDT ne . and _AENDT eq . and
    (input(substr(RCENDTC, 1, 10), ??is8601da.)>VAX102DT or (upcase(ONGNXVIS)='Y' and AENDT eq .) ) then
do;
  AENDT=VAX102DT;
  ADURN=AENDT-ASTDT+1;
  ADURU="DAYS";
end;
```

```
attrib TRTP length=$100. label="Planned Treatment"
  TRTA length=$100. label="Actual Treatment"
  TRTPN label="Planned Treatment (N)"
  TRTAN label="Actual Treatment (N)";
```

```
TRTA=TRT01A;
TRTAN=TRT01AN;
TRTP=TRT01P;
TRTPN=TRT01PN;
if (VAX101='Placebo' and CETPTREF="VACCINATION 1") or (VAX102='Placebo' and
CETPTREF="VACCINATION 2") then do;
  TRTA='Placebo';TRTAN=9;
end;
if (VAX102=' ' and CETPTREF="VACCINATION 2") then do;
  TRTA=' ';TRTAN=.;
end;
if adurn=. then aduru="";
```

FDA-CBER-2022-5812-0072078

```

drop cedir;
run;

proc sort data=adcevd;
  where cetptref ne ";
  by usubjid CETPTREF CETERM ASEV;
run;

*****;
* Specification 6 Output final data      *;
*****;
%let g_adcevd_vars=%str(TRTA TRTAN TRTP TRTPN SRCDOM SRCSEQ CEGRPID CELNKGRP CETERM
CELLT CEDECOD CEPTCD CEBODSYS CEBDSYCD
CECAT CECAT CEPRESP CEOCCUR CESTAT CEREASND ASEV CESEV CELOC CELAT CESTDTC
CEENDTC CESTDY CEENDY CEENRTPT CEENTPT CETPT CETPTREF
EXDOSE EXTRT EXDOSU EXSTDTC EXENDTC ASTDT AENDT ADURU ADURN KNOWVFL EVENTFL);

data datvprot.adcevd(label='Diary and CRF Event Analysis Dataset');
  retain &g_adsl_vars &g_adcevd_vars;
  set adcevd(keep=&g_adsl_vars &g_adcevd_vars);

  label ADURN='Analysis Duration (N)'
        ADURU='Analysis Duration Units'
        ASEV='Analysis Severity/Intensity'
        KNOWVFL='Known Value Flag'
        EVENTFL='Event Value Flag'
        ASTDT='Analysis Start Date'
        AENDT='Analysis End Date'
        SRCSEQ='Source Sequence Number'
        SRCDOM='Source Data';

run;

proc printto;
run;

```