

```

options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
* macro to save output and log to appropriate areas ;
%_mprintto;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106326;
%put NOTE: Client Protocol ID : ZRHM-PK-05-JP;
%put NOTE: Program Name : d_2ADAE.sas;
%put NOTE: Purpose : create ADAE dataset;
%put NOTE: ;
%put NOTE: Input Data : STDLIB.ADAE SDTM.AE SDTM.SUPPAE;
%put NOTE: Output : ADAM.ADAE;
%put NOTE: Macros Called : _MPRINTTO _MTOTPER _MPERALL;
%put NOTE: ;
%put NOTE: Programmed by : cvn_kbooth;
%put NOTE: Creation Date : 2014-04-13;
%put NOTE: SAS Version : 9.3;
%put NOTE: ;
%put NOTE: == Latest Run
=====;
%put NOTE: Run by : &sysuserid;
%put NOTE: Date/Time :
%sysfunc(putn(%sysfunc(date()),e8601da.))T%sysfunc(putn(%sysfunc(time()),
e86011z.));
%put NOTE: ;
%put NOTE: == Modification History
=====;
%put NOTE: Date Initials No. Reason;
%put NOTE: 05Aug2014 KB 1) Added EXNOTRFL and NICOGR2 variables
to keep;
%put NOTE: 05Aug2014 KB 2) Amended reference to AEACP ;
%put NOTE: 05Aug2014 KB 3) Amended reference to AEENRF;
%put NOTE: 05Aug2014 KB 4) Removed formats for AETRTEM, AEEXPEC
and AEEXPEC1;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

*=====;
* START OF PROGRAM CODE ;
*=====;
*****;
* bring in ADSL ;
*****;

data adsl;
set adam.adsl;
keep studyid usubjid subjid: siteid age sex: race height weightb1
bmi ucpdgr1 ucpdgr1n nicogr1 nicogr1n NICOGR2 NICOGR2N targr1 targr1n /*
1) KB 05Aug2014 */

```

```

enrlfl scrffl complfl fupfl saffl randfl exfl enfl trt:
tr01: tr02: dthfl lvisdt lvisday dtestdt ptestdt dsreas: analgr:
EXNOTRFL; /* 1) KB 05Aug2014 */
run;

*****;
* bring in SUPPAE ;
*****;

proc transpose data = sdtm.suppaе out=suppaе(drop = _:);
  var qval;
  by usubjid idvarval;
  id qnam;
  idlabel qlabel;
run;

data suppaе2(drop = idvarval);
  set suppaе;
  aeseq = input(idvarval,best.);
run;
*****;
* bring in AE ;
*****;

data ae;
  merge sdtm.ae suppaе2;
  by usubjid aeseq;
  format astdt aendtc date9. aedurn aesern 8. aeduru $10. aeongfl $2.
/*aetrtem aeexpec aeexpec1 $200.*/; /* 4) KB 05Aug2014 */
  * dates;
  if length(aestdct) gt 10 then astdt =
input(scan(aestdct,1,'T'),ymmdd10.);
  else if length(aestdct) = 10 then astdt = input(aestdct,ymmdd10.);

  if length(aeendtc) gt 10 then aendtc =
input(scan(aeendtc,1,'T'),ymmdd10.);
  else if length(aeendtc) = 10 then aendtc = input(aeendtc,ymmdd10.);

  * duration;
  if not missing(astdt) and not missing(aendtc) then do;
    aedurn = aendtc - astdt + 1;
    aeduru = 'Days';
  end;

  * ongoing flag ;
  if /*aeenrf*/AEENRTPT = 'ONGOING' then aeongfl = 'Y'; /* 3) KB
05Aug2014 */
  else aeongfl = 'N';

  * serious;
  if aeser = 'N' then aesern = 0;
  else if aeser = 'Y' then aesern = 1;

  * severity ;

```

```

    if aesev = 'MILD' then aesevn = 1;
    else if aesev = 'MODERATE' then aesevn = 2;
    else if aesev = 'SEVERE' then aesevn = 3;
    else if not missing(aesev) then put 'Warn' 'ing: check severity as
unexpected response: ' usubjid = aesev = ;
    if missing(aesev) then aesevn = 3; * SAP section 12.6.4.2.1;

    * causality;
    if aerelsp = 'NOT RELATED' then aerelspn = 0;
    else if aerelsp = 'RELATED' then aerelspn = 1;
    else if not missing(aerelsp) then put 'Warn' 'ing: check other
related text as unexpected: ' usubjid = aerelsp = ;

    * action ;
    if /*aeacp*/AEACNP1 = 'NONE' then /*aeacpn*/AEACNP1N = 5; /* 2) KB
05Aug2014 */
    else if /*aeacp*/AEACNP1 = 'PRODUCT USE INTERRUPTED' then
/*aeacpn*/AEACNP1N = 1; /* 2) KB 05Aug2014 */
    else if /*aeacp*/AEACNP1 = 'PRODUCT USE REDUCED' then
/*aeacpn*/AEACNP1N = 2; /* 2) KB 05Aug2014 */
    else if /*aeacp*/AEACNP1 = 'PRODUCT USE STOPPED' then
/*aeacpn*/AEACNP1N = 3; /* 2) KB 05Aug2014 */
    else if /*aeacp*/AEACNP1 = 'NOT APPLICABLE' then /*aeacpn*/AEACNP1N
= 4; /* 2) KB 05Aug2014 */
    else if not missing(/*aeacp*/AEACNP1) then put 'Warn' 'ing: check
action as unexpected response: ' usubjid = /*aeacp*/ AEACNP1= ; /* 2) KB
05Aug2014 */

    * outcome;
    if aeout = 'DEATH RELATED TO ADVERSE EVENT' then aeoutn = 1;
    else if aeout = 'NOT RECOVERED/NOT RESOLVED' then aeoutn = 2;
    else if aeout = 'RECOVERED/RESOLVED' then aeoutn = 3;
    else if aeout = 'RECOVERED/RESOLVED WITH SEQUELAE' then aeoutn = 4;
    else if aeout = 'RECOVERING/RESOLVING' then aeoutn = 5;
    else if aeout = 'UNKNOWN' then aeoutn = 6;
    else if not missing(aeout) then put 'Warn' 'ing: check outcome as
unexpected response: ' usubjid = aeout =;

    * some supp variables missing will need to include when run off dry
run;
    keep usubjid aeseq aeexpec aeexpec1 aespid aetrtem aeterm aedecod
aebodsys aebdsydc aellt: aept: aehlt: aehlg: aesoc:
    aestdtc astdt aeendtc aendt aestdy aeendy aedurn aedurru
aeongfl aeser: aesev: aerel /*aeacp*/AEACNP1: aeout: /* 2) KB
05Aug2014 */
    aescong aesdisab aesdth aeshosp aeslife aecontrt aeacnoth
epoch aerelsp aerelspn;
run;

/* Obtain AEREL1 & AEREL1N */
data aerel1;
    set sdtm.suppae(where=(qnam='AEREL1'));

    idvarval2=input(idvarval,best.);

```

```

        rename idvarval2=aeseq
              qval=aerel1;

    if qval in ('Y', 'RELATED') then do;
        aerel1n=1;
    end;
    else do;
        aerel1n=0;
    end;

        keep usubjid idvarval2 qval aerel1n;
run;

proc sort data=aerel1;
    by usubjid aeseq;
run;

/* Obtain AEREL2 & AEREL2N */
data aerel2;
    set sdtm.suppaе (where=(qnam='AEREL2'));

        idvarval2=input(idvarval,best.);

        rename idvarval2=aeseq
              qval=aerel2;

    if qval in ('Y', 'RELATED') then do;
        aerel2n=1;
    end;
    else do;
        aerel2n=0;
    end;

        keep usubjid idvarval2 qval aerel2n;
run;

proc sort data=aerel2;
    by usubjid aeseq;
run;

proc sort data=ae;
    by usubjid aeseq;
run;

data ae2;
    merge aerel1 aerel2 ae;
    by usubjid aeseq;
run;

*****;
* Combine ADSL and data *;
*****;
proc sort data=ae2;

```

```

        by usubjid;
run;

/* macro to allocate period and actual treatment information from ADSL */
%_mtotper;

data slae(drop = astdtm dsreas:);
    merge adsl ae2(in=a);
    by usubjid;

    format aperiod trtan trtpn astday aenday 8. trta trtp $40. aperiodc
$8.
        anyae fl anl01fl anl02fl anl03fl anl04fl anl05fl trtemfl prefl
$2.;
    astday = astdt - trtsdt + 1;
    aenday = aendt - trtsdt + 1;
    astdtm = .; * for macro below to work ;

    /* Pre-treatment flag */
    if aerelspn=1 and ((astdt < dtestdt) or (astdt < ptestdt)) then
prefl='y';

    * flag anyae ;
    if a then anyae fl = 'Y';
    else do;
        anyae fl = 'N';
    end;

    * product emergent same or after first product use;
    if not missing(astdt) and not missing (dtestdt) and astdt ge
dtestdt then trtemfl = 'Y';
    else if anyae fl = 'Y' then trtemfl = 'N';

    * flag analysis record flags;
    * exposure emergent or study related prior to exposure;
    if trtemfl='Y' or prefl='Y' then anl01fl='Y';
    if saffl ne 'Y' and anl01fl = 'Y' then put 'USER WARN' 'ING:
subject not in safety population but event to be presented in summaries?
' usubjid = aeterm = aestdte = ;

    * AEs leading to withdrawal;
    if trtemfl = 'Y' then do;
        if upcase(dsreas) = 'ADVERSE EVENTS' | upcase(dsreas) =
'ADVERSE EVENTS' then do;
            if aeacnoth in ('SUBJECT DISQUALIFIED' 'DISCONTINUED')
then anl02fl = 'Y';
            else put 'USER WARN' 'ING: check withdrawal information
as inconsistent: ' usubjid = dsreas =;
        end;

        * AEs where product stopped or discontinued;
        if (aerel2n = 1 or aerel1n = 1) and /*aeacpn*/AEACNP1N in (1
2 3) then anl03fl = 'Y'; /* 2) KB 05Aug2014 */

```

```

        * AEs requiring conmeds;
        if aecontrt='Y' then anl04fl = 'Y';

        * Action taken = other;
        if not missing(aeacnoth) then anl05fl = 'Y';
    end; * exposure emergent AEs for summaries;

    * declare full and partial dates for deriving period;

    if astday in (0 1) then aperiod=1;
    else if astday in (2 3) then aperiod=2;
    %_mperall(dvar1 = astdtm, dvar2 = astdt);

    if not missing(aperiod) then do;
        aperiodc = 'Period ' || put(aperiod,1.);
    end;

run;

*****;
* find highest severity for each pt term ;
*****;
* sorted by anl01fl - all AEs exposure emergent or study related prior to
exposure;
proc sort data = slae;
    by trtemfl usubjid aedecod descending aesevn;
run;

data ae3;
    set slae;
    by trtemfl usubjid aedecod descending aesevn;
    format aoccpifl $2.;
    if trtemfl = 'Y' and first.aedecod then aoccpifl = 'Y';
run;

*****;
* find max related for each pt term ;
*****;

proc sort data = ae3;
    by trtemfl usubjid aedecod descending aerelln;
run;

data ae4;
    set ae3;
    by trtemfl usubjid aedecod descending aerelln;
    format aoccpf1 $2.;
    if trtemfl = 'Y' and first.aedecod then aoccpf1 = 'Y';
run;

proc sort data=ae4;
    by usubjid aedecod descending aerelln;
run;

```

```

*****;
* create output dataset ;
*****;

options replace;

data adae;
    set stdlib.adae ae4;
    label aperiodc = 'Period (C)';

    drop trt01p: trt02p: trt01a: trt02a: dtestdt ptestdt;
run;

proc sort data = adae out = adam.adae(label= 'Adverse Event Analysis
Dataset');
    by usubjid aeterm aeecod aebodsys aestdtc;
run;

options noreplace;

%_scramble(set=adae, id=usubjid subjid subjidn age sex sexc sexn race
dthfl height weightbl bmi ucpdgr1 ucpdgrln nicogr1
        nicogrln targr1 targrln analgr1 analgrln, dates=trtsdtm
trtsdt trtsday trtedtm trtedt trteday tr01sdt tr01stm tr01sdtm tr01edt
tr01etm tr01edtm
        tr02sdt tr02stm tr02sdtm tr02edt tr02etm tr02edtm,
nullc=trtp trta trtseqp trtseqa aerel2 aerel1 /*aeacp*/AEACNP1 aeacnoth
aeexpec aeexpec1 trtstmf tr01stmf tr02stmf, nulln=aerel2n
/*aeacpn*/AEACNP1N /* 2) KB 05Aug2014 */
        aerel1n trtseqpn trtseqan trtpn trtan, nullcc=13,
nullnc=7);

proc printto; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```