

```

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 : 000000106324;
%put NOTE: Client Protocol ID   : ZRHR-REXC-03-EU;
%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;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_smulholl;
%put NOTE: Creation Date        : 2013-10-18;
%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: 29Nov2013  SM          1)  NONE removed from AETERM if no
events reported;
%put NOTE:                                     2)  Remove test code in adae
data step;
%put NOTE:                                     3)  Correct codes for AECPN to
match codelist;
%put NOTE:                                     4)  Amend code for TRTEMFL to
include N;
%put NOTE: 29Apr2014  KB          5)  Amended ADURU to AEDURU;
%put NOTE: 29Apr2014  KB          6)  Amended ANL01FL;
%put NOTE: 29Apr2014  KB          7)  Amended ANL03FL;
%put NOTE: 29Apr2014  KB          8)  Amended AERELN;
%put NOTE: 29Apr2014  KB          9)  Added EPOCH to keep statement;
%put NOTE: 29Apr2014  KB          10) Amended setting of shell;
%put NOTE: 29Apr2014  KB          11) Added PREFL to code;
%put NOTE: 29Apr2014  KB          12) Removed format from AESEQ;
%put NOTE: 29Apr2014  KB          13) Added code to only populate AEDURN
and AEDURU if start and end dates are not blank;
%put NOTE: 12May2014  KB          14) Amended sorting key variables;
%put NOTE: 14May2014  KB          15) Amended ANL02FL;
%put NOTE: 14May2014  KB          16) Added AETRTEM to keep;
%put NOTE: 14May2014  KB          17) Amended APERIOD and APERIODC to
populate for AEs;
%put NOTE: 14May2014  KB          18) Amended sorting for AOCCPIFL and
AOCCPRFL;

```

```

%put NOTE: 26Jun2014    JM          19) Use AEENRTPT to replace AEENRF for
ONGONING information due to AEENRF was removed from MH dataset;
%put NOTE: 27Jul2014    KB          20) Added EXNOTRFL;
%put NOTE: 27Jul2014    KB          21) Removed AETRTEM;
%put NOTE: 12Sep2014    KB          22) Added FASFL & PPROTFL to keep from
ADSL;
%put NOTE: ;
%put NOTE: ;
%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 ucpdgrln nicogr1 nicogrln targr1 targrln cobl
        enrfl scrffl complfl fupfl saffl randfl exfl EXNOTRFL enfl
trt: tr01: dthfl lvisdt lvisday dtestdt dsreas: FASFL PPROTFL; /* 20) KB
27Jul2014 */ /* 22) KB 12Sep2014 */
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;
    /* format aeseq 8.; */ /* 12) KB 29Apr2014 */
    aeseq = input(idvarval,best.);
run;
*****;
* bring in AE ;
*****;

/* 15) START KB 14May2014 */
DATA DS;
    SET SDTM.DS (WHERE= (INDEX (DSDECOD, 'ADVERSE')) );

```

```

        FLAG=1;
        KEEP USUBJID FLAG;
RUN;

DATA AE1;
    MERGE SDTM.AE DS;
    BY USUBJID;
RUN;

PROC SORT DATA=AE1;
    BY USUBJID AESEQ;
RUN;
/* 15) END KB 14May2014 */

data ae;
    merge /*sdtm.ae*/AE1 suppa2; /* 15) KB 14May2014 */
    by usubjid aeseq;
    format astdt aendt date9. aedurn aesern 8. /*aduru*/AEDURU $10.
aeongfl $2.; /* 5) KB 29Apr2014 */
    * dates;
    if length(aestdtc) gt 10 then astdt =
input(scan(aestdtc,1,'T'),yymmdd10.);
    else if length(aestdtc) = 10 then astdt = input(aestdtc,yymmdd10.);

    if length(aeendtc) gt 10 then aendt =
input(scan(aeendtc,1,'T'),yymmdd10.);
    else if length(aeendtc) = 10 then aendt = input(aeendtc,yymmdd10.);

    * duration;
    if not missing(astdt) and not missing(aendt) then DO; /* 13) KB
29Apr2014 */
        aedurn = aendt - astdt + 1;
        /*aduru*/AEDURU = 'Days'; /* 5) KB 29Apr2014 */
    END; /* 13) KB 29Apr2014 */

    * ongoing flag ;
/*    if aeenrf = 'ONGOING' then aeongfl = 'Y';*/
    if aeenrtpt = 'ONGOING' and missing(AEENDTC) then aeongfl = 'Y'; /*
19) JM 26JUN2014*/
    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 aere1 = 'N' then aere1n = 0;*/ /* 8) KB 29Apr2014 */
/*else*/ if aere1 in ('Y' /*'R'/'RELATED') then aere1n = 1; /* 8)
KB 29Apr2014 */
      ELSE AERE1N=0; /* 8) KB 29Apr2014 */
      /*else*/ if not missing(aere1) AND MISSING(AERE1N) then put 'Warn'
'ing: check causality as unexpected response: ' usubjid = aere1 = ; /*
8) KB 29Apr2014 */
      if aere1sp = 'NOT RELATED' then aere1spn = 0;
      else if aere1sp = 'RELATED' then aere1spn = 1;
      else if not missing(aere1sp) then put 'Warn' 'ing: check other
related text as unexpected: ' usubjid = aere1sp = ;

      * action ;
      if /*aeacp*/ aeacnpl = 'NONE' then /*aeacpn*/ aeacnpln = 5; /* 20)
JM 26JUN2014*/
      else if /*aeacp*/ aeacnpl = 'PRODUCT USE INTERRUPTED' then
/*aeacpn*/ aeacnpln = 1; /* 20) JM 26JUN2014*/
      else if /*aeacp*/ aeacnpl = 'PRODUCT USE REDUCED' then /*aeacpn*/
aeacnpln = 2;*3; /* 3) SM 29Nov2013 */ /* 20) JM 26JUN2014*/
      else if /*aeacp*/ aeacnpl = 'PRODUCT USE STOPPED' then /*aeacpn*/
aeacnpln = 3;*2; /* 3) SM 29Nov2013 */ /* 20) JM 26JUN2014*/
      else if /*aeacp*/ aeacnpl = 'NOT APPLICABLE' then /*aeacpn*/
aeacnpln = 4; /* 20) JM 26JUN2014*/
      else if not missing(/*aeacp*/ aeacnpl) then put 'Warn' 'ing: check
action as unexpected response: ' usubjid = /*aeacp*/ aeacnpl = ;/* 20) JM
26JUN2014*/

      * 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 aesp1d aeterm aedecod aebodsys aebdsydc
aellt: aept: aehlt: aehlg: aesoc:
      aestdtc astdt aeendtc aendt aestdy aeendy aedurn
/*aduru*/AEDURU aeongfl aeser: aesev: aere1: /*aeacp*/ aeacnpl: aeout: /*
5) KB 29Apr2014 */ /* 20) JM 26JUN2014*/
      aescong aesdisab aesdth aeshosp aeslife aecontrt aeacnoth
EPOCH FLAG AETRTEM; /* 9) KB 29Apr2014 */ /* 15) KB 14May2014 */ /* 16)
KB 14May2014 */
run;

*****;
* Combine ADSL and data *;

```

```

*****;

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

data slae(drop = trt01: tr01: astdtm dsreas: dtestdt);
    merge adsl ae(in=a);
    by usubjid;

    format aperiod trtan trtpn astday aenday 8. trta trtp $40. aperiodc
$10.
        anyae fl anl01fl anl02fl anl03fl anl04fl anl05fl trtemfl PREFL
$2.; /* 11) KB 29Apr2014 */
    astday = astdt - trtsdt + 1;
    aenday = aendt - trtsdt + 1;
    astdtm = .; * for macro below to work ;

    * flag anyae ;
    if a then anyae fl = 'Y';
    else do;
        anyae fl = 'N';
        *aeterm = 'NONE'; /* 1) SM 29Nov2013 */
    end;

    /* Pre-treatment flag*/
    IF AERELSPN=1 AND (ASTDT < DTESTDT) THEN PREFL='Y'; /* 11) KB
29Apr2014 */

    * 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 ANYAEFL = 'Y' THEN TRTEMFL = 'N'; /* 4) SM 29Nov2013 */

    * flag analysis record flags;
    /*if trtemfl = 'Y' then do;*/ /* 6) KB 29Apr2014 */
        * exposure emergent or study related prior to exposure;
/*
        if not missing(astdt) and ((astdt < dtestdt and aerelspn = 1)
or (astdt ge dtestdt)) then anl01fl = 'Y';*/ /* 6) KB 29Apr2014 */
        IF TRTEMFL='Y' OR PREFL='Y' THEN ANL01FL='Y'; /* 6) KB
29Apr2014 */
        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 = aestdtc = ;

    IF TRTEMFL='Y' THEN DO; /* 6) KB 29Apr2014 */
        * AEs leading to withdrawal;
/*
        if upcase(dsreas) = 'ADVERSE EVENTS' | upcase(dsreasp) =
'ADVERSE EVENTS' then do;*/
            IF FLAG=1 THEN DO; /* 15) KB 14May2014 */
                if aeacnoth in ('SUBJECT DISQUALIFIED' 'DISCONTINUED')
then anl02fl = 'Y';*/
                IF INDEX(AEACNOTH,'DISQUALIFIED') OR
INDEX(AEACNOTH,'DISCONTINUED') THEN ANL02FL='Y'; /* 15) KB 14May2014 */

```

```

/*          else put 'USER WARN' 'ING: check withdrawal information
as inconsistent: ' usubjid = dsreas =;*/ /* 15) KB 14May2014 */
end;

      * AEs where product stopped or discontinued;
      if AEREL='RELATED' AND /*aeacpn*/ aeacnpln in (/*2 3*/1 2 3)
then anl03fl = 'Y'; /* 7) KB 29Apr2014 */ /* 20) JM 26JUN2014*/

      * 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;
/* 17) START KB 14May2014 */
      IF ANYAEFL='Y' THEN DO;
        %_mperall(dvar1 = astdtm, dvar2 = astdt);

        aperiodc = 'Period ' || put(aperiod,1.);
      END;
/* 17) END KB 14May2014 */

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 ASTDT; /* 18) KB
14May2014 */
run;

data ae2(DROP=FLAG AETRTEM); /* 15) KB 14May2014 */ /* 21) KB 27Jul2014
*/
  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 = ae2;
  by trtemfl usubjid aeDecod descending aeReln ASTDT; /* 18) KB
14May2014 */
run;

data ae3;

```

```

        set ae2;
        by trtemfl usubjid aeDecod descending aereln;
        format aoccprfl $2.;
        if trtemfl = 'Y' and first.aeDecod then aoccprfl = 'Y';
run;

/* 10) START KB 29Apr2014 */
PROC SORT DATA=AE3;
    BY USUBJID AETERM AEDECOD AEBODSYS AESTDTC;
RUN;
/* 10) END KB 29Apr2014 */

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

options replace;

data adae;
    set stdlib.adae /*slae*/AE3; /* 10) KB 29Apr2014 */
    label aperiodc = 'Period (C)';
    *rename trtp=prodp trta=proda trtpn=prodpn trtan=prodan; /* 2) SM
29Nov2013 */
run;

proc sort data = adae out = adam.adae(label= 'Adverse Event Analysis
Dataset');
/*    by usubjid aeterm aeDecod aebodsys aestdtk;*/
    BY USUBJID AETERM AEDECOD AEBODSYS AESTDTC AESPID; /* 14) KB
12May2014 */
run;

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

```