

```

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_1ADSL.sas;
%put NOTE: Purpose : create ADSL dataset;
%put NOTE: ;
%put NOTE: Input Data :STDLIB.ADSL SDTM.DM SDTM.SUPPDM SDTM.VS
SDTM.SU SDTM.FA;
%put NOTE: SDTM.EX SDTM.DX SDTM.DS SDTM.VS
SDTM.SV SDTM.IE SDTM.LB;
%put NOTE: Output : ADAM.ADSL;
%put NOTE: Macros Called : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by : cvn_smulholl;
%put NOTE: Creation Date : 2013-09-10;
%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) Drop D_3 from final data;
%put NOTE: 28Apr2014 KB 2) Amended WEIGHTBL and BMI;
%put NOTE: 28Apr2014 KB 3) Amended derivation of UCPDGR1;
%put NOTE: 28Apr2014 KB 4) Amended NICOGR1 to match other
studies;
%put NOTE: 28Apr2014 KB 5) Amended DISCDT;
%put NOTE: 28Apr2014 KB 6) Amended EXFL and ENFL;
%put NOTE: 28Apr2014 KB 7) Removed SDTM.DS from merge;
%put NOTE: 28Apr2014 KB 8) Added EPOCH to condition for FUPFL;
%put NOTE: 28Apr2014 KB 9) Amended condition for FASFL to check
for RANDFL;
%put NOTE: 28Apr2014 KB 10) Amended FASREAS for no biomarker
data;
%put NOTE: 28Apr2014 KB 11) Amended SAFFL to check for
performed data;
%put NOTE: 28Apr2014 KB 12) Added DMRANDNO;
%put NOTE: 28Apr2014 KB 13) Amended TRT01A to check for ENFL;
%put NOTE: 28Apr2014 KB 14) Amended TRTEDAY derivation;
%put NOTE: 28Apr2014 KB 15) Added TRTSTMF TRTETMF TR01STMF
TR01ETMF;
%put NOTE: 28Apr2014 KB 16) Amended pull out of smoking history
data due to updated SDTM;

```

[illegible]

```

%put NOTE: ;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

*=====;
* START OF PROGRAM CODE ;
*=====;

*****
*****;
* pick up demography data ;
* create numeric variables and code according to value level metadata in
ADaM specs ;
*****
*****;
data dm;
    set sdtm.dm;
    format subjidn sexn racen armcdn ethnictn 8. brthdt date9. sexc $20.
        rfstdt date9. scrffl $2.;
    subjidn = input(subjid,best.);
    brthdt = input(brthdtc,ymmdd10.);
    if sex = 'M' then do;
        sexc = 'Male';
        sexn = 1;
    end;
    else if sex = 'F' then do;
        sexc = 'Female';
        sexn = 2;
    end;
    if race = 'WHITE' then racen = 1;
    else put 'Warn' 'ing: Check race for SDTM controlled terms: '
race=; * while code developing ;
    if ethnic = 'CAUCASIAN' then ethnictn = 1;
    else if ethnic = 'NOT CAUCASIAN' then ethnictn = 2;
    else if ethnic = 'JAPANESE' then ethnictn = 3;
    else if ethnic = 'NOT JAPANESE' then ethnictn = 4;
    else if ethnic = 'HISPANIC' then ethnictn = 5;
    else if ethnic = 'NOT HISPANIC' then ethnictn = 6;
    else if (missing(ethnic) and armcd ne 'SCRNFAIL') or not
missing(ethnic) then put 'USER WARN' 'ING: Check Ethnicity for controlled
terms: ' usubjid = ethnic = ;

    *study reference dates ;
    if not missing(rfstdtc) and length(rfstdtc) gt 10 then rfstdt =
/*datepart(*input(SCAN(rfstdtc,/*is8601dt.*/1,'T'),YMMDD10.)/*)*/; /*
32) KB 21Jun2014 */
    else if not missing(rfstdtc) and length(rfstdtc) = 10 then rfstdt =
input(rfstdtc,ymmdd10.);

```

```

* numeric arm coding variables;
* study specific;
if armcd = 'THS 2.2' then armcdn = 1;
else if armcd = 'CC' then armcdn = 2;
else if armcd = 'SMABST' then armcdn = 3;
else if armcd = 'SCRNFAIL' then armcdn = 4;
ELSE IF ARMCD='NOTASSGN' THEN ARMCDN=20; /* 31) KB 21Jun2014 */
ELSE PUT "USER WA" "RNING: Check ARMCDs " ARMCD=; /* 31) KB 21Jun2014
*/

```

```

* screen failure flag ;
if armcd not in ('THS 2.2' 'CC' 'SMABST') then scrffl='Y';
else scrffl='N';

```

```

keep studyid usubjid subjid subjidn siteid age ageu brthdtc brthdt
sex sexc sexn race racen ethnic: country arm
armcd armcdn rfstdt dthfl scrffl;
run;

```

```

*****;
* pick up if race=other for specify details ;
* macro to check if SUPPDM dataset exists ;
*****;
%macro _mcheckdata(dset1=, dset2=);

```

```

%let no = 0; * does suppdm exist ;
%let racecheck = 0; * does race=OTHER exist ;

```

```

data _null_;
set dm;
if race = 'OTHER' then call symput('racecheck','1');
run;

```

```

%if &racecheck. = 1 %then %do; * race=OTHER exists ;

```

```

data _null_;
if (exist("&dset1..", "SDTM")) then do;
put "USER NO" "TE &dset1 does exist.";
dsid = open("&dset1.", "I"); *remove condition if not searching
for any records;
rc = fetch(dsid);
if rc = -1 then do;
put "USER WARN" "ING &dset1. has no data.";
call symput('no',0);
end;
else do;
call symput('no',1);
end;
rc = close(dsid);
return;
end;
else do;
put "USER WARN" "ING &dset1. does not exist.";
call symput('no',0);

```

```

        stop;
        end;
run;

* if data has indicated OTHER is present in race then check to make sure
SUPPDM exists;
* write a note to the log if not ;

%if &racecheck. = 1 and &no. ne 1 %then %put "USER WARN" "ING: Data
indicates OTHER race present but &dset1. does not exist";

* if SUPP file does exist ;

%if &no. = 1 %then %do;
    proc transpose data = sdtm.&dset1. out = &dset1.a;
        by usubjid idvarval;
        id qnam;
        var qval;
    run;

    proc sort data = &dset1.a;
        by usubjid;
    run;

    data &dset2.a;
        merge &dset2. &dset1.a;
        by usubjid;
        if index(race,'OTHER') and missing(raceoth) then put 'USER
WARN' 'ING: Other information missing - query with DM: ' usubjid=;
    run;
%end;
%end;
* if SUPP file does not exist;
%if &no. = 0 %then %do;

%if &racecheck. = 0 %then %put 'USER NOTE: race = OTHER does not exist';

    data &dset2.a;
        set &dset2.;
    run;
%end;

%mend _mcheckdata;
%_mcheckdata(dset1=suppdm, dset2=dm);

/* 12) START KB 28Apr2014 */
DATA RANDNO;
    SET SDTM.SUPPDM(WHERE=(QNAM='DMRANDNO'));
/*    FORMAT DMRANDNO $10.;*/ /* 41) KB 24Jun2014 */

```

```

        IF QNAM='DMRANDNO' THEN DMRANDNO=COMPRESS(QVAL);

        KEEP USUBJID DMRANDNO;
RUN;

DATA DMA2;
    MERGE DMA(IN=A) RANDNO;
    BY USUBJID;
    IF A;
RUN;
/* 12) END KB 28Apr2014 */

*****
*****;
* Bring in vital signs data for weight height and BMI ;
* baseline is usually screening - check for study specific requirements ;
*****
*****;
/* 2) START KB 28Apr2014 */
DATA WEIGHT;
    SET SDTM.VS(WHERE=(VSTESTCD IN ('HEIGHT' 'WEIGHT')));

    IF VSTESTCD='WEIGHT' AND VISIT NE 'DAY -2' THEN DELETE;
RUN;
/* 2) END KB 28Apr2014 */

proc transpose data = /*sdtm.vs*/WEIGHT/*(where=(vsblfl='Y' and vstestcd
in ('WEIGHT' 'HEIGHT' 'BMI'))*/ out = tvs (drop=_: rename = (weight =
weightbl)); /* 2) KB 28Apr2014 */
    var vsstresn;
    by usubjid;
    id vstestcd;
run;

*****;
* produce derived classification parameters for BMI ;
*****;
data vs2;
    set tvs;
    format bmigr1 BMIUNR1 $40. bmigrln 8. BMI 8.1 HEIGHT WEIGHTBL
BEST.; /* 2) KB 28Apr2014 */ /* 24) KB 07May2014 */

    BMI=PUT(WEIGHTBL/((HEIGHT/100)**2),8.1); /* 2) KB 28Apr2014 */
    BMIUNR=WEIGHTBL/((HEIGHT/100)**2); /* 24) KB 07May2014 */

    if 0 < bmi < 18.5 then do;
        bmigr1 = 'Underweight';
        bmigrln = 1;
    end;
    else if 18.5 <= bmi < 25 then do;
        bmigr1 = 'Normal weight';
        bmigrln = 2;
    end;
    else if 25 <= bmi < 30 then do;

```

```

        bmigrl = 'Overweight';
        bmigrln = 3;
    end;
    else if bmi >= 30 then do;
        bmigrl = 'Obese';
        bmigrln = 4;
    end;
    else if not missing(bmi) then put 'USER WARN' 'ING BMI
unclassified: ' usubjid= bmi=;

/* 24) START KB 07May2014 */
    IF 0 < BMIUNR < 18.5 THEN DO;
        BMIUNR1 = 'Underweight';
    END;
    ELSE IF 18.5 <= BMIUNR < 25 THEN DO;
        BMIUNR1 = 'Normal weight';
    END;
    ELSE IF 25 <= BMIUNR < 30 THEN DO;
        BMIUNR1 = 'Overweight';
    END;
    ELSE IF BMIUNR >= 30 THEN DO;
        BMIUNR1 = 'Obese';
    END;

/*      IF BMIGR1 NE BMIUNR1 THEN PUT "USER WARN" "ING: BMIs don't match "
USUBJID=;*/ /* 26) KB 27May2014 */

    DROP BMIUNR1 BMIUNR;
/* 24) END KB 07May2014 */

run;

*****;
* add to DMA ;
*****;

data dm2;
    merge /*dma*/DMA2(in = a) vs2(in = b); /* 12) KB 28Apr2014 */
    by usubjid;
    if (a and not b) and armcd ne 'SCRNFAIL' then put 'USER WARN' 'ING
baseline weight, height and BMI not available: ' usubjid= weightbl=
height= bmi=;
run;

*****
*****;
* Bring in daily cigarette consumption for classification in summary and
analysis;
* This may originate by questionnaire or from randomisation ;
* Check study aCRF and specifications for more information ;
*****
*****;

```

```

data fa(keep = usubjid ucpdgr:);
    set sdtm.fa(where=(faobj = /*'Smoking History'*/'SMOKING HISTORY'
and fatestd = 'SMOKHIST' and epoch=/*'ADMI'*/'SCREENING')); /* 16) KB
28Apr2014 */ /* 43) KB 24Jun2014 */
    format ucpdgr1 $40. ucpdgrln 8.;
    if index(/*fastresc*/FAORRES,'<10') then do; /* 3) KB 28Apr2014 */
        ucpdgrln = 1;
        ucpdgr1 = '<10 cig/day';
    end;
    else if /*fastresc*/FAORRES = '10 to 19' then do; /* 3) KB
28Apr2014 */
        ucpdgrln = 2;
        ucpdgr1 = '10-19 cig/day';
    end;
    else if index(/*fastresc*/FAORRES,'>19') then do; /* 3) KB
28Apr2014 */
        ucpdgrln = 3;
        ucpdgr1 = '>19 cig/day';
    end;
    else put 'USER WARN' 'ING unable to classify daily cigarette
consumption: ' usubjid= /*fastresc*/FAORRES=; /* 3) KB 28Apr2014 */
    output; /* only keep usual
daily cig consumption;
run;

*****;
* add to DM2 ;
*****;

data dm3;
    merge dm2(in = a) fa(in = b);
    by usubjid;
    if (a and not b) and armcd ne 'SCRNFAIL' then put 'USER WARN' 'ING
baseline daily cigarette classification not available: ' usubjid=;
run;

*****
*****;
* Bring in baseline cig nicotine yield for classification in summary and
analysis;
* for baseline only - check specifications ;
* Check study aCRF and specifications for more information ;
*****
*****;

data fal(keep = usubjid nico:);
    set sdtm.fa(where = (fatestcd = 'NYIELD' and epoch = 'ADMI'));
    format nicobl best8. nicogrln 8. nicogr1 $20.;
    nicobl = fastresn;
    if not missing(nicobl) and nicobl le 0.6 then do;
        nicogr1 = '<= 0.6 mg';
        nicogrln = 1;
    end;
    else if 0.6 < nicobl <= 1 then do;
/*
        nicogr1 = '> 0.6 mg to <= 1 mg';*/

```



```

/*          NICOGR1 = '> 0.6 mg - <= 1.0 mg'; *//* 4) KB 28Apr2014 */
          NICOGR1 = '> 0.6 - 1.0 mg'; /* 38) KB 23Jun2014 */
          nicogrln = 2;
        end;
      else if nicobl > 1 then do;
/*          nicogr1 = '> 1 mg';*/
          nicogr1 = '> 1.0 mg'; /* 4) KB 28Apr2014 */
          nicogrln = 3;
        end;
      else put 'USER WARN' 'ING unable to classify nicotine yield at
baseline consumption: ' usubjid= nicobl=;
      output;                                * only keep nicotine
yield information;
run;

*****
*****;
* Bring in baseline tar yield for classification in summary and analysis;
* for baseline only - check specifications ;
* Check study aCRF and specifications for more information ;
*****
*****;
data fa2(keep = usubjid tar:);
  set sdtm.fa(where = (fatestcd = 'TYIELD' and epoch = 'ADMI'));
  format tarbl targrln 8. targr1 $20.;
  tarbl = fastresn;
  if 1 le FLOOR(tarbl) le 5 then do; /* 35) KB 23Jun2014 */
    targr1 = '1-5 mg';
    targrln = 1;
  end;
  else if 6 le FLOOR(tarbl) le 8 then do; /* 35) KB 23Jun2014 */
    targr1 = '6-8 mg';
    targrln = 2;
  end;
  else if 9 le FLOOR(tarbl) le 10 then do; /* 35) KB 23Jun2014
*/
    targr1 = '9-10 mg';
    targrln = 3;
  end;
  else if tarbl gt 10 then do;
    targr1 = '>10 mg';
    targrln = 4;
  end;
  else put 'USER WARN' 'ING unable to classify tar yield at baseline:
' usubjid= tarbl=;
  output;                                * only keep tar yield
information;
run;

*****
*****;
* Bring in baseline tar yield for classification in summary and analysis;
* for baseline only - check specifications ;
* Check study aCRF and specifications for more information ;

```

```

*****
*****;
data fa3(keep = usubjid cobl);
    set sdtm.fa(where = (fatestcd = 'COYIELD' and epoch = 'ADMI'));
    format cobl 8.;
    cobl = fastresn;
    output;                                * only keep CO level
at baseline information;
run;

*****
*****;
* Bring in if device test performed;
* Check study aCRF and specifications for more information ;
*****
*****;
proc sort data = sdtm.dx out = dx(where = (not missing(dxstdtc)));
    by usubjid dxstdtc dxendtc;
run;

data fa4(keep = usubjid dtest:);
    merge sdtm.fa(where = (fatestcd = 'PERFORM' and faorres='Yes'))
          dx(where = (dxstdy lt 1) keep = usubjid dxstdtc dxstdy);
    by usubjid;
    if first.usubjid;
    format dtestdtm datetime13. dtestdt date9. dtesttm time5.;
    if length(dxstdtc) gt 10 then do;
/*          dtestdtm = input(dxstdtc,e8601dt.);*/

DTESTDTM=DHMS(INPUT(SCAN(DXSTDTC,1,'T'),YYMMDD10.),HOUR(INPUT(SCAN(DXSTDTC,2,'T'),TIME5.)),MINUTE(INPUT(SCAN(DXSTDTC,2,'T'),TIME5.)),0); /* 32) KB
21Jun2014 */
        dtesttm      = timepart(dtestdtm);
        dtestdt      = datepart(dtestdtm);
    end;
    else if not missing(dxstdtc) then dtestdt =
input(scan(dxstdtc,1,'T'),yyymmdd10.);
run;

*****;
* add to DM3 ;
*****;

data dm4;
    merge dm3(in = a) fa1(in = b) fa2(in = c) fa3(in = d) fa4;
    by usubjid;
    if armcd ne 'SCRNFAIL' then do;
        if (a and not b) then put 'USER WARN' 'ING baseline nicotine
yield classification not available: ' usubjid=;
        if (a and not c) then put 'USER WARN' 'ING baseline tar yield
classification not available: ' usubjid=;
        if (a and not d) then put 'USER WARN' 'ING baseline CO level
not available: ' usubjid=;

```

```

        end;
run;

*****;
* bring in exposure data ;
*****;

*****;
* SA arm information - no device or product used ;
* so using visit dates with nominal times from protocol ;
*****;

proc sort data = sdtm.sv(where = (epoch = 'PRODUCT USE CONFINEMENT'))
out=sa;
    by usubjid svstdtc;
run;

data saarm(keep = usubjid exstdtc exendtc exstdy ARMCD); /* 18) KB
28Apr2014 */
    merge sa sdtm.dm(keep = usubjid armcd);
    by usubjid;
    format exstdtc exendtc /*$40.*/$16.; /* 33) KB 21Jun2014 */
    exstdtc = trim(svstdtc) || 'T06:30'; * nominal start time for SA
arm ;
    exendtc = trim(svstdtc) || 'T23:00'; * nominal end time for SA
arm ;
    rename svstdy = exstdy;
    if armcd = 'SMABST' then output; * only keep SA arm subjects ;
run;

* device admin data;
data dxa;
    set sdtm.dx(where = (not missing(dxstdtc) and dxstdy ge 1)); *
don't include device test data ;
run;

proc sort data = dxa;
    by usubjid;
run;

* cc admin data ;
data ex;
    set sdtm.ex(where=(exstdy ge 1 and not missing(exstdtc)));* don't
include admission data for cc data ;
run;

proc sort data = ex;
    by usubjid;
run;

* combine for full product admin set ;
data exp;
    set dxa(rename = (dxstdtc = exstdtc dxendtc = exendtc
dxstdy=exstdy)) ex;

```

```

        by usubjid;
        keep usubjid exstdtc exendtc exstdy;
run;

data expall;
    merge exp(in=exdx) saarm (in=none);
    by usubjid;
    if exdx and none then put 'USER WARN' 'ING: check subject
treatment/exposure dates as in both SA and exposed: ' usubjid = ;
run;

* sort in date order ;
proc sort data = expall;
    by usubjid exstdy exstdtc exendtc;
run;

data ex2s(keep = usubjid trts: tr01s:) ex2e(keep = usubjid trte: tr01e:);
    set expall;
    by usubjid;
    format trtsdtm trtedtm tr01sdtm tr01edtm datetime13. trtsdt trtedt
tr01sdt tr01edt date9.
    tr01stm tr01etm time5. TRTSTMF TR01STMF TRTETMF TR01ETMF $1.; /*
23) KB 07May2014 */
    if first.usubjid then do; * first administration of abstinence,
device or cc for day 1;
        if not missing(exstdtc) then do;
/*            trtsdtm = input(exstdtc,is8601dt.);*/

TRTSDTM=DHMS (INPUT (SCAN (EXSTDTC,1,'T'),YYMMDD10.),HOUR (INPUT (SCAN (EXSTDTC
,2,'T'),TIME5.)),MINUTE (INPUT (SCAN (EXSTDTC,2,'T'),TIME5.)),0); /* 32) KB
21Jun2014 */

            trtsdt = datepart(trtsdtm);
            tr01sdtm = trtsdtm;
            tr01sdt = trtsdt;
            tr01stm = timepart(trtsdtm);

/* 15) START KB 28Apr2014 */
            IF ARMCD='SMABST' THEN DO;
                TRTSTMF='H';
                TR01STMF='H';
            END;
/* 15) END KB 28Apr2014 */
        end;
        output ex2s;
    end;
    if last.usubjid then do; *last administration or day of abstinence
;
        if not missing(exendtc) then do;
/*            trtedtm = input(exendtc,is8601dt.);*/

TRTEDTM=DHMS (INPUT (SCAN (EXENDTC,1,'T'),YYMMDD10.),HOUR (INPUT (SCAN (EXENDTC
,2,'T'),TIME5.)),MINUTE (INPUT (SCAN (EXENDTC,2,'T'),TIME5.)),0); /* 32) KB
21Jun2014 */

```

```

        trtedt = datepart(trtedtm);
        tr01edtm = trtedtm;
        tr01edt = trtedt;
        tr01etm = timepart(trtedtm);

/* 15) START KB 28Apr2014 */
        IF ARMCD='SMABST' THEN DO;
            TRTETMF='H';
            TR01ETMF='H';
        END;
/* 15) END KB 28Apr2014 */
        end;
        output ex2e;
    end;
run;

*****;
* determine actual treatments ;
*****;
*smoking abstinence;
proc sort data=saarm;
    by usubjid exstdy exstdtc;
run;

proc transpose data = saarm out = tsa(drop = _) prefix = s;
    var exstdtc;
    by usubjid;
    id exstdy;
run;

*device;
proc sort data = dx;
    by usubjid dxstdy dxstdtc;
run;

data dx2;
    set dx(where = (not missing(dxstdtc)));
    by usubjid dxstdy;
    if first.dxstdy;
run;

proc transpose data = dx2(where=(not missing(dxstdy))) out = tdx2(drop =
_) prefix = d;
    var dxstdtc;
    by usubjid;
    id dxstdy;
run;

*conventional cigarettes;

proc sort data = ex;
    by usubjid exstdy exstdtc;
run;

```

```

data ex2;
    set ex(where = (not missing(exstdtc)));
    by usubjid exstdy;
    if first.exstdy;
run;

proc transpose data = ex2 out=tex2(drop = _) prefix=e;
    var exstdtc;
    by usubjid;
    id exstdy;
run;

data treats(drop = devcount cccount sacount i);
    merge tex2 tdx2 tsa;
    by usubjid;
    * check compliance to product;
    array a [5] d1-d5;
    array b [5] e1-e5;
    array c [5] s1-s5;
    devcount=0;
    cccount=0;
    sacount=0;
    do i=1 to 5;
        if not missing(a[i]) then devcount+1;
        if not missing(b[i]) then cccount+1;
        if not missing(c[i]) then sacount+1;
    end;
    if devcount ne 0 and (cccount ne 0 or sacount ne 0) then put 'USER
WARN' 'ING: THS device used but other products exposure' usubjid=;
    if cccount ne 0 and (devcount ne 0 or sacount ne 0) then put 'USER
WARN' 'ING: CC used but other products exposure' usubjid=;
    if sacount ne 0 and (cccount ne 0 or devcount ne 0) then put 'USER
WARN' 'ING: Abstinence used but other products exposure' usubjid=;
run;

*****;
* add to DM4 ;
*****;

data dm5(drop = rfstdt /*s_3*/); /* 28) KB 04Jun2014 */
    merge dm4(in = a) ex2s(in = b) ex2e(in = c) treats;
    by usubjid;

    if a and not b and scrffl='N' then put 'USER WARN' 'ING exposure
data not available: ' usubjid=;
    format trtsday trteday dtestday 8.;
    * derive exposure start and end days using start of study reference
(Day 1) ;
    if not missing(rfstdt) then do;
        trtsday = trtsdt - rfstdt + 1;
        trteday = trtedt - /*trtsdt*/RFSTDT + 1; /* 14) KB 28Apr2014
*/

```

```

end;
dtestday = dtestdt - trtsdt + 1;

DROP SCRFFL; /* 19) KB 28Apr2014 */
run;

*****;
* introduce DS data for flags and dates for informed consent etc ;
*****;
*all subjects with disposition information;
proc sort data = sdtm.ds out = allds(keep = usubjid) nodupkey;
    by usubjid;
run;

* informed consent;
data infcons;
    merge sdtm.ds(where = (dscat = 'PROTOCOL MILESTONE' and dsdecod =
'INFORMED CONSENT OBTAINED' and dsterm = 'MAIN INFORMED CONSENT') in = a)
        allds;
    by usubjid;
    format /*icfdtc $20.*/ICFDTM DATETIME13. icfdt date9.; /* 42) KB
24Jun2014 */
/*    icfdtc = trim(dsstdtc);*/

ICFDTM=DHMS(INPUT(SCAN(DSSTDTC,1,'T'),YMMDD10.),HOUR(INPUT(SCAN(DSSTDTC,
2,'T'),TIME5.)),MINUTE(INPUT(SCAN(DSSTDTC,2,'T'),TIME5.)),0); /* 42) KB
24Jun2014 */
    icfdt = input(dsstdtc,yymmdd10.);
    keep usubjid /*icfdtc*/ICFDTM icfdt; /* 42) KB 24Jun2014 */
run;

*****;
* to be enrolled, the subject must comply with all eligibility ;
* criteria (no observations in IE) and have performed a device ;
* test and be willing and able to use the device ;
*****;

* eligibility;
proc sort data = sdtm.ie out = ie(keep = usubjid) nodupkey;
    by usubjid;
run;

* device test;
proc sort data=sdtm.fa(where = (/*(fatestcd = 'PERFORM' and faorres =
'Yes') or*/ (fatestcd = 'WILLABL' and faorres = /*'Yes'*/'No')))) /* 20)
KB 28Apr2014 */
    out=device(keep = usubjid) nodupkey;
    by usubjid;
run;

/* 6) START KB 28Apr2014 */
DATA PERF;
    SET SDTM.FA(WHERE=(FATESTCD IN ('PERFORM')));

```

```

        KEEP USUBJID FAORRES FATESTCD;
RUN;

PROC TRANSPOSE DATA=PERF OUT=PERF2(DROP=_:);
    BY USUBJID;
    VAR FAORRES;
    ID FATESTCD;
RUN;

DATA PERF3;
    SET PERF2;

    IF PERFORM NE 'Yes' THEN DELETE;

    KEEP USUBJID;
RUN;
/* 6) END KB 28Apr2014 */

data enr;
    merge /*sdtm.ds(where = (dscat = 'DISPOSITION EVENT' and dsterm =
'DISCONTINUED FROM ENROLLMENT') in = a)*/ /* 7) KB 28Apr2014 */
        allods ie(in=b) device(in=/*c*/D) PERF3(IN=C) ; /* 6) KB
28Apr2014 */ /* 20) KB 28Apr2014 */
    by usubjid;
    format enrfl $2. ENFL EXFL SCRFFL EXNOTRFL $2.; /* 6) KB 28Apr2014
*/ /* 23) KB 07May2014 */ /* 37) KB 23Jun2014 */
    if /*a or*/ b or not c then enrfl = 'N';      * discontinued,
invalid inclusion or exclusion or no valid device test; /* 7) KB
28Apr2014 */
    else enrfl = 'Y'; * enrolled flag - all IE critieria not met at
admission or screening;

/* 19) START KB 28Apr2014 */
    IF ENRLFL='Y' THEN SCRFFL='N';
    ELSE IF ENRLFL='N' THEN SCRFFL='Y';
/* 19) END KB 28Apr2014 */

/* 6) START KB 28Apr2014 */
    IF NOT C THEN EXFL='N';
    ELSE EXFL='Y';
    IF B THEN ENFL='N';
    ELSE ENFL='Y';

    IF ENFL='Y' THEN EXFL='N';
    IF SCRFFL='Y' THEN DO;
        EXFL='N';
        ENFL='N';
    END;
/* 6) END KB 28Apr2014 */

/* 20) START KB 28Apr2014 */
    IF NOT C AND NOT B AND SCRFFL NE 'Y' THEN PUT "USER WA" "RNING:
Product test not done but all criteria passed, check if deviation
recorded " USUBJID=; /* 44) KB 11Sep2014 */

```



```

        IF D AND NOT B THEN PUT "USER WA" "RNING: WILLABL not equal to yes
but no record in IE " USUBJID=;
/* 20) END KB 28Apr2014 */

        IF NOT C THEN EXNOTRFL = "N"; /* 37) KB 23Jun2014 */
        ELSE EXNOTRFL = "Y"; /* 37) KB 23Jun2014 */

        keep usubjid enrfl EXFL ENFL SCRFFL EXNOTRFL; /* 6) KB 28Apr2014
*/ /* 19) KB 28Apr2014 */ /* 37) KB 23Jun2014 */
run;

* completion details;
data suplds;
    set sdtm.suplds(where=(qnam = 'OTHER'));
    dsseq = input(idvarval,best.);
    format dsreasp $200.;
    dsreasp = trim(propcase(qval,'/'));
    keep usubjid dsseq dsreasp;
run;

/* 5) START KB 28Apr2014 */
DATA SUPPDS2;
    MERGE SUPPDS(IN=A) SDTM.DS;
    BY USUBJID DSSEQ;
    IF A;
RUN;

PROC SORT DATA=SUPPDS2(WHERE=(DSCAT='DISPOSITION EVENT'));
    BY USUBJID DSSTDTC;
RUN;

DATA SUPPDS3;
    SET SUPPDS2;
    BY USUBJID DSSTDTC;

    IF FIRST.USUBJID AND FIRST.DSSTDTC THEN OUTPUT;
RUN;

DATA DSTEST2;
    SET SDTM.DS(WHERE =(DSCAT = 'DISPOSITION EVENT')) ;
RUN;

PROC SORT DATA=DSTEST2;
    BY USUBJID DSSTDTC;
RUN;

DATA DSTEST3;
    SET DSTEST2;
    BY USUBJID DSSTDTC;

    IF FIRST.USUBJID THEN OUTPUT;
RUN;

PROC SORT DATA=DSTEST3;

```

```

        BY USUBJID DSSEQ;
RUN;
/* 5) END KB 28Apr2014 */

data compa lastobs(/*where =(dscat = 'DISPOSITION EVENT')*/ keep =
usubjid dsdecod dscat rename = (dsdecod = extrads)); /* 5) KB 28Apr2014
*/
        merge /*sdtm.ds*/DSTEST3 /*suppds*/SUPPDS3; /* 5) KB 28Apr2014 */
        by usubjid dsseq;
        if last.usubjid then output lastobs; * final followup;
        output compa;
run;

data comp;
        merge compa(where = (dscat = 'DISPOSITION EVENT'))
                allds lastobs(drop = dscat)
SDTM.DS (WHERE=(FUEPOCH="FOLLOWUP") KEEP = USUBJID EPOCH RENAME=(EPOCH =
FUEPOCH) IN=FU); /* 40) KB 23Jun2014 */
        by usubjid;
        if first.usubjid;
        format complfl fupfl $2. discdt date9. dsreas dsreasp $200.;
        if dsterm = 'COMPLETED' then complfl = 'Y'; * completion of study
- all parts;
        else complfl = 'N';
        if dsterm='LOST TO FOLLOW-UP' then fupfl='N'; * completion of
follow-up;
        else IF /*EPOCH='FOLLOWUP'*/ FU THEN fupfl='Y'; /* 8) KB 28Apr2014
*/ /* 40) KB 23Jun2014 */
        ELSE IF NOT FU THEN FUPFL='N'; /* 8) KB 28Apr2014 */ /* 40) KB
23Jun2014 */
        * date of completion\discontinuation;
        discdt = input(dsstdtc, yymmdd10.);
        *reason for discontinuation;
        if dsdecod='SCREEN FAILURE' then dsreas =
trim(propcase(dsterm, '/'));
        else if dsdecod ne 'COMPLETED' then dsreas =
trim(propcase(dsterm, '/'));
        if dsterm ne extrads and dsdecod ne 'SCREEN FAILURE' then dsreasp =
trim(propcase(extrads, '/'));
        keep usubjid complfl fupfl discdt dsreas; ;
run;

* randomisation details ;
data rand(keep = usubjid rand:);
        merge sdtm.ds(where = (dscat = 'PROTOCOL MILESTONE' and dsdecod =
'RANDOMIZED') in = a)
                allds;
        by usubjid;
        format randfl $2. randdtm datetime13. randdt date9.;
        if a then randfl = 'Y';
        else randfl = 'N';
        if randfl='Y' then do;
/*                randtm = input(dsstdtc,e8601dt.);*/

```

```

RANDDTM=DHMS (INPUT (SCAN (DSSTDTC,1,'T'),YYMMDD10.),HOUR (INPUT (SCAN (DSSTDTC
,2,'T'),TIME5.)),MINUTE (INPUT (SCAN (DSSTDTC,2,'T'),TIME5.)),0); /* 32) KB
21Jun2014 */
    randdt = datepart(randdtm);
end;
run;

* combine flags ;
data flag;
    merge enr infcons comp rand;
    by usubjid;
run;

data dates1(keep = usubjid icf01:) dates2(keep = usubjid icf02:);
    set sdtm.ds(where = (dscat = 'OTHER EVENT' and dsdecod = 'INFORMED
CONSENT OBTAINED'));
    by usubjid;
    format icf01dtc icf02dtc $20. icf01dt icf02dt date9. ;
    if dsterm = /*'INFORMED CONSENT OBTAINED FOR
TRANSCRIPTOMICS'*/'INFORMED CONSENT FOR TRANSCRIPTOMICS' then do; /* 34)
KB 21Jun2014 */
        icf01dtc = trim(dsstdtc);
        icf01dt = input(dsstdtc,yyymmdd10.);
        output dates1;
    end;
    if dsterm = 'INFORMED CONSENT FOR BIOMARKERS' then do;
        icf02dtc = trim(dsstdtc);
        icf02dt = input(dsstdtc,yyymmdd10.);
        output dates2;
    end;
run;

*****;
* add to DM5 ;
*****;

data dm6;
    merge dm5(in = a) flag(in = b) dates1(in = c) dates2(in = d);
    by usubjid;
    * subject not in DS data;
    if a and not b and not missing(armcd) then put 'USER WARN' 'ING
flagging data not available: ' usubjid=;
    * informed consent missing for trns and bio samples;
/*    if a and not (c or d) and not missing(randdt) then put 'USER WARN'
'ING: informed consent date missing: ' usubjid = ;*/ /* 45) KB 11Sep2014
*/
    format icfday 8./* enfl exfl $2.*/; /* 6) KB 28Apr2014 */
    icfday=icfdt - trtsdt + 1;
    * flag if enrolled but not randomised ;
/*    if enrlfl = 'Y' and randfl = 'N' then enfl = 'Y';*/ /* 6) KB
28Apr2014 */
/*    else enfl = 'N';*/ /* 6) KB 28Apr2014 */
    * flag if exposed but not randomised ;

```

```

/*    if not missing(dtestdt) and randfl = 'N' then exfl = 'Y';*/ /* 6)
KB 28Apr2014 */
/*    else exfl = 'N'; */ /* 6) KB 28Apr2014 */

    IF RANDFL='Y' AND ENFL='Y' THEN ENFL='N'; /* 6) KB 28Apr2014 */
    IF RANDFL='Y' AND ENRLFL='N' THEN ENRLFL='Y'; /* 36) KB 23Jun2014 */
    IF RANDFL = "Y" AND EXNOTRFL = "Y" THEN EXNOTRFL = "N"; /* 37) KB
23Jun2014 */

run;

* safety flagging ;
data dm6a;
    merge dm6 /*device(in=device)*/PERF3(IN=PERF); * trial period ;
/* 11) KB 28Apr2014 */
    by usubjid;
    format saffl $2. safreas /*$40.*/$50.; /* 41) KB 24Jun2014 */
    if not missing(icfdt) and /*device*/PERF then saffl = 'Y'; /* 11)
KB 28Apr2014 */
    else saffl = 'N';
    if saffl = 'N' then do;
        if missing(icfdt) and not /*device*/PERF then safreas = 'No
exposure to THS 2.2 and No informed consent'; /* 11) KB 28Apr2014 */
        else if missing(/*icfdtc*/ICFDTM) then safreas = 'No informed
consent'; /* 42) KB 24Jun2014 */
        else if not /*device*/PERF then safreas = 'No exposure to THS
2.2'; /* 11) KB 28Apr2014 */
    end;
run;

* treatments;
data dm6b;
    set dm6a;
    format trt01pn trt01an 8. trt01p trt01a $40.;

    * planned treatments;
    if randfl = 'Y' then do; * randomised subjects ;
    if armcd = 'THS 2.2' then do;
        trt01pn = 1;
        trt01p = armcd;
    end;
    else if armcd = 'CC' then do;
        trt01pn = 2;
        trt01p = armcd;
    end;
    else if armcd = 'SMABST' then do;
        trt01pn = 3;
        trt01p = 'SA';
    end;
    end;
    else if armcd = 'ENRNORAN' or enfl = 'Y' then do;
        trt01pn = 97;
        trt01p = 'Enrolled not randomized';
    end;
end;

```

```

else if exfl = 'Y' then do;
    trt01pn = 98;
    trt01p = 'Exposed not randomized';
end;
else if armcd = 'SCRNFAIL' then do;
    trt01pn = 99;
    trt01p = 'Screen failure';
end;
* actual treatments ;
if randfl = 'Y' then do; * randomised subjects ;
if armcd = 'THS 2.2' then do;
    if not missing(d1) then do;
        trt01an = 1;
        trt01a = armcd;
    end;
    else put 'USER WARN' 'ING: Randomised to THS but no exposure
date/time: ' usubjid = ;
    end;
else if armcd = 'CC' then do;
    if not missing(e1) then do;
        trt01an = 2;
        trt01a = armcd;
    end;
    else put 'USER WARN' 'ING: Randomised to CC but no exposure
date/time: ' usubjid = ;
    end;
else if armcd = 'SMABST' then do;
    if not missing(s1) then do;
        trt01an = 3;
        trt01a = 'SA';
    end;
    else put 'USER WARN' 'ING: Randomised to SA but exposure
date/time recorded: ' usubjid = ;
    end;
end;
/* 13) START KB 28Apr2014 */
ELSE IF ENFL='Y' THEN DO;
    TRT01AN = 97;
    TRT01A = 'Enrolled not randomized';
END;
/* 13) END KB 28Apr2014 */
else if exfl = 'Y' then do;
    trt01an = 98;
    trt01a = 'Exposed not randomized';
end;
else if scrffl='Y' then do;
    trt01an = 99;
    trt01a = 'Screen failure';
end;
run;

* bring in DV data for PP population;
data dv;

```

```

        set sdtm.dv(where=(dvcat='MAJOR'));
        keep usubjid;
run;

proc sort data=dv nodupkey;
    by usubjid;
run;

* lab data for Biomarkers ;
proc sort data = sdtm.lb(where = (lbcat not in ('HAEMATOLOGY' 'CLINICAL
CHEMISTRY' 'SEROLOGY' 'URINALYSIS' 'COTININE SCREENING')
    and missing(lbstat))) out=lb(keep = usubjid) nodupkey;
    by usubjid;
run;

/* 25) START KB 27May2014 */
/* Importing banned meds sheet and checking CM */
/* Checking for banned medications in SDTM.CM */
PROC IMPORT

DATAFILE="/cvn/projects/prj/data/000000106324/source/bannedmeds.xlsx"
    OUT=WORK.MEDLIST
    REPLACE
    DBMS=XLSX;
    RANGE="A1:A68";
    SHEET='CYP2A6';
    GETNAMES=NO;
RUN;

PROC SORT DATA =MEDLIST(RENAME = (A = CHECK)) NODUPKEY;
    BY CHECK;
RUN;

DATA MEDLIST2A;
    SET MEDLIST;
    DUMMY=1;

    IF COMPRESS(CHECK,, 'KW')='Inhibitor' THEN DELETE;
    IF COMPRESS(CHECK,, 'KW')='Inducer' THEN DELETE;
    IF COMPRESS(CHECK,, 'KW')='Substrate' THEN DELETE;
RUN;

PROC TRANSPOSE DATA=MEDLIST2A OUT=MEDLIST2;
    BY DUMMY;
    VAR CHECK;
RUN;

DATA CMMEDS;
    SET SDTM.CM(WHERE=(CMENDY GE -16 OR /*CMENRF*/CMENRTPT='ONGOING'))
    KEEP=USUBJID CMTRT CMDECOD CMENDY /*CMENRF*/CMENRTPT); /* 30) KB
21Jun2014 */

    CHECK=LOWCASE(CMDECOD);

```

```

        DUMMY=1;
RUN;

PROC SORT DATA=CMMEDS;
    BY CHECK;
RUN;

DATA CMMEDS2 (DROP = CHECK);
    MERGE CMMEDS (IN = CM) MEDLIST2 (IN = LIST);
    BY DUMMY;
    IF CM;
    FORMAT CRIT1FL $2. CRIT1 $50.;
    LENGTH CHECK2 $200;
    CRIT1 = 'Affects CYP2A6';
    ARRAY A [66] COL1 - COL66;
    DO I=1 TO 66;
        IF INDEX(COMPRESS(CHECK,, 'KA'), COMPRESS(A[I],, 'KA')) AND
        (*CMENRF*/ CMENRTPT EQ 'ONGOING' OR /*-15*/-16<=CMENDY<=-2) THEN DO; /*
29) KB 05Jun2014 */ /* 30) KB 21Jun2014 */
            CHECK2=COMPRESS(A[I],, 'KA');
            IF SCAN(CHECK,1) NE CHECK2 THEN DELETE;
            CRIT1FL = 'Y';
        OUTPUT;
    END;
END;
RUN;

PROC SORT DATA=CMMEDS2;
    BY USUBJID;
RUN;

PROC IMPORT

DATAFILE="/cvn/projects/prj/data/000000106324/source/bannedmeds.xlsx"
    OUT=WORK.MEDLISTB
    REPLACE
    DBMS=XLSX;
    RANGE="A1:A68";
    SHEET='CYP1A2';
    GETNAMES=NO;
RUN;

PROC SORT DATA =MEDLISTB (RENAME = (A = CHECK)) NODUPKEY;
    BY CHECK;
RUN;

DATA MEDLIST2B;
    SET MEDLISTB;
    DUMMY=1;

    IF COMPRESS(CHECK,, 'KW')='Inhibitor' THEN DELETE;
    IF COMPRESS(CHECK,, 'KW')='Inducer' THEN DELETE;
    IF COMPRESS(CHECK,, 'KW')='Substrate' THEN DELETE;

```

```

        IF COMPRESS(CHECK,, 'KW')='CYP 1A2 and CYP 2A6: Substrated,
inhibitors, inducers' THEN DELETE;
RUN;

PROC TRANSPOSE DATA=MEDLIST2B OUT=MEDLIST3;
    BY DUMMY;
    VAR CHECK;
RUN;

DATA CMMEDS2B/*(DROP = CHECK)*/;
    MERGE CMMEDS(IN = CM) MEDLIST3(IN = LIST);
    BY DUMMY;
    IF CM;
    FORMAT CRIT2FL $2. CRIT2 $50.;
    LENGTH CHECK2 $200;
    CRIT2 = 'Affects CYP1A2';

    ARRAY A [66] COL1 - COL66;
    DO I=1 TO 66;
        IF INDEX(COMPRESS(CHECK,, 'KA'),LOWCASE(COMPRESS(A[I],, 'KA')) AND
(/*CMENRF*/CMENRTPT EQ 'ONGOING' OR /*-15*/-16<=CMENDY<=-2) THEN DO; /*
29) KB 05Jun2014 */ /* 30) KB 21Jun2014 */
            CHECK2=COMPRESS(A[I],, 'KA');
            CRIT2FL = 'Y';
            OUTPUT;
        END;
    END;
RUN;

PROC SORT DATA=CMMEDS2B;
    BY USUBJID;
RUN;

DATA CMMEDS3;
    SET CMMEDS2(DROP=COL:) CMMEDS2B(DROP=COL:);
RUN;

PROC SORT DATA=CMMEDS3(WHERE=(CRIT1FL='Y' OR CRIT2FL='Y') KEEP=USUBJID
CRIT1FL CRIT2FL);
    BY USUBJID;
RUN;
/* 25) END KB 27May2014 */

data dm6c(drop = e1-e5 /*D_3*/ d1-d5 s1-s5 iflg rflg fasflg eflg dvflg
lflg CRIT1FL CRIT2FL CMFLG); /* 1) SM 29Nov2013 */ /* 17) KB 28Apr2014
*/ /* 25) KB 27May2014 */
    merge dm6b dv(in = dv) lb(in = c) CMMEDS3(IN=CM); /* 25) KB
27May2014 */
    by usubjid;
    format fasfl pprotfl $2. fasreas ppreas $200.;
    if /*safll = 'Y' and not missing(randdt)*/RANDFL='Y' and c then do;
* safety and randomised with biomarker data; /* 9) KB 28Apr2014 */
        if ((armcd = 'THS 2.2' and not missing(d1)) or (armcd = 'CC'
and not missing(e1))) or armcd = 'SMABST' then fasfl = 'Y';

```



```

end;
else do;
    fasfl = 'N';
    if missing(icfdat) then iflg = 'Did not given informed
consent.';
    else if /*missing(randdt)*/RANDFL='N' then rflg = 'Was not
randomized'; /* 9) KB 28Apr2014 */
    else if missing(d1) and missing(e1) and trt01p ne 'SA' then
eflg = 'Did not have post-randomization smoking event.';
    else if not c then lflag = /*'Did not have valid BoExp
measurement.'*/'Did not have any valid non-safety post-randomization
assessments'; /* 10) KB 28Apr2014 */
    fasreas = compbl(trim(iflg)|| ' ' || trim(rflg)|| ' '
||trim(eflg)) || ' ' || trim(lflag);
end;

FASREAS=STRIP(FASREAS); /* 22) KB 07May2014 */

if fasfl = 'Y' and not dv AND NOT CM then pprotfl = 'Y'; * subject
has deviation; /* 25) KB 27May2014 */
else do;
    pprotfl = 'N';
    if fasfl = 'N' then fasflg = 'Not in FAS';
    if dv then dvflg = 'Has major protocol deviation';
    IF CM THEN CMFLG='Use of medications known to affect CYP2A6 or
CYP1A2.'; /* 25) KB 27May2014 */
    ppreas = catx(' ', trim(fasflg), trim(dvflg), TRIM(CMFLG));
/* 25) KB 27May2014 */
end;

run;

*****;
* introduce SV data for dates of visits ;
*****;

data sv;
    set sdtm.sv;
    by usubjid;
    if last.usubjid;
    format lvisdt date9. lvisdtc $20. lvisit $40.;
    lvisdtc = trim(/*svstdtc*/SVENDTDC); /* 27) KB 27May2014 */
    lvisdt = input(scan(lvisdtc,1,'T'),yymmdd10.);
    lvisit = trim(propcase(visit,'/'));
    keep usubjid lvis; ;
run;

*****;
* add to DM6c ;
*****;

data dm7(DROP=D_3); /* 21) KB 07May2014 */
    merge dm6c sv;
    by usubjid;

```

```

        format lvisday 8.;
        lvisday = lvisdt - trtsdt + 1;
run;
*****;
* create output dataset ;
*****;

options replace;

data adsl;
    set stdlib.adsl dm7;
run;

proc sort data = adsl out = adam.adsl(label= 'Subject Level Analysis
Dataset');
    by usubjid;
run;

options noreplace;

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

```