

```
/*=====
=====*
```

```
| Covance Study Number   : 000000106343          |
| Program Name           : t_device.sas          |
| Purpose                 : Summary of THS 2.2 Menthol Device Events - Safety Population      |
| Input Data              : ADAM.ADSL, ADAM.ADDE   |
| Output Data             : tlfds.T_15_02_06_15_01 & RTF      |
| Macros Called           : m_printto, m_logchk    |
| Originally Performed by : kpothuri              |
| Date                    : 04MAY2015             |
```

```
|
|=====
=====|
```

```
| Modification History          |
|-----|
| Modified by                   :                   |
| Modification Date             :                   |
| Modification Description      :                   |
```

```
+=====
=====*/
```

```
options notes source source2 nofullstimer validvarname=upcase missing=' ';
```

```
ods _all_ close;
```

```
ods listing;
```

```
%m_printto(route=YES);
```

```
*=====;
```

```

* START OF PROGRAM CODE                                ;

*=====;

proc datasets lib=work nolist memtype=data kill; quit;

%let TFL_Part=%scan(&_SASPROGRAMFILE,-3,%str(/));

/* Standard - leave this */

data _null_;

    tmp("&TFL_Part");

    if tmp not in ("dev" "qc") then call symput("TFL_Part", "prod");

    call symput('TFLpath', compress("&_SASPROGRAMFILE", ""));

    call symput('TFLprg', reverse(scan(strip(reverse(compress("&_SASPROGRAMFILE", ""))),1,"/")));

run;

options missing=' ';

data de;

    set adam.adde;

    output;

    trta="Total";

    trtan=1;

    output;

run;

```

*N - counts - pre-randomization;

data adsl;

 set adam.adsl;

 output;

 trt01a="Total";

 trt01an=1;

 output;

run;

proc sql noprint;

 create table big_n as

 select trt01an, trt01a, count(distinct(usubjid))

 from adsl

 where SAFBFL="Y"

 group by trt01an, trt01a;

quit;

data big_n;

 set big_n;

 if trt01a="THSm2.2" then trt01a="THS";

 if trt01a="Product Test" then trt01a="ProdT";

run;

data _null_;

 set big_n;

 call symput ("B_" || compress(trt01a), put(_TEMG001,8.));

run;

%put &B_SA &B_THS &B_mCC &B_ProdT &B_Total;

*N - counts - after randomization;

```
proc sql noprint;
```

```
    create table big_n_aft as
```

```
    select trt01an, trt01a, count(distinct(usubjid))
```

```
    from adsl
```

```
    where SAFAFL="Y"
```

```
    group by trt01an, trt01a;
```

```
quit;
```

```
data big_n_aft;
```

```
    set big_n_aft;
```

```
    if trt01a="THSm2.2" then trt01a="THS";
```

```
    if trt01a="Product Test" then trt01a="ProdT";
```

```
run;
```

```
data _null_;
```

```
    set big_n_aft;
```

```
    call symput ("A_" || compress(trt01a), put(_TEMG001,8.));
```

```
run;
```

```
%put &A_THS;
```

```
%macro pd (pd=, tab=, name=, where=);
```

```
proc sql noprint;
```

```
    *device events;
```

```
create table dev as  
  
select trta, trtan, count(distinct(usubjid))  
  
from de  
  
where AEREL ne "" and &where  
  
group by trta, trtan;
```

```
create table dev_ev as  
  
select trta, trtan, count(usubjid)  
  
from de  
  
where AEREL ne "" and &where  
  
group by trta, trtan;
```

```
*related/not related to ae;  
  
create table ae as  
  
select trta, trtan, AEREL, count(distinct(usubjid))  
  
from de  
  
where AEREL ne "" and &where  
  
group by trta, trtan, AEREL;
```

```
create table ae_ev as  
  
select trta, trtan, AEREL, count(usubjid)  
  
from de  
  
where AEREL ne "" and &where  
  
group by trta, trtan, AEREL;
```

```
quit;
```

```
*product test is missing - pre-rand;

data dev;

    set dev;

    output;

    if trta="SA" and trta ne "Product Test" then do;

        trta="Product Test";

        _TEMG001=0;

        trtan=.;

        output;

    end;

run;
```

```
data dev_ev;

    set dev_ev;

    output;

    if trta="SA" and trta ne "Product Test" then do;

        trta="Product Test";

        _TEMG001=0;

        trtan=.;

        output;

    end;

run;
```

```
*device events;
```

```
proc transpose data=dev out=t_dev prefix=N_;
```

```
    id trta;
```

```
    var _temg001;
```

```
run;
```

```
proc transpose data=dev_ev out=t_dev_ev prefix=E_;
```

```
    id trta;
```

```
    var _temg001;
```

```
run;
```

```
data de_1;
```

```
length label $200;
```

```
    merge t_dev t_dev_ev;
```

```
    by _name_;
```

```
    ord=1;
```

```
    label="Device Events";
```

```
run;
```

```
*related/not related to ae;
```

```
proc sort data=ae; by aere1; run;
```

```
proc transpose data=ae out=t_ae prefix=N_;
```

```
    id trta;
```

```
    var _temg001;
```

```
    by aere1;
```

```
run;
```

```
proc sort data=ae_ev; by aere1; run;
```

```
proc transpose data=ae_ev out=t_ae_ev prefix=E_;
```

```
    id trta;
```

```
    var _temg001;
```

```
    by aere1;
```

```
run;
```

```
data de_2;
```

```
length label $200;
```

```
    merge t_ae t_ae_ev;
```

```
    by _name_ aere1;
```

```
    if aere1 = "NOT RELATED" then do; ord=3; label="  Not Related to AE"; output; end;
```

```
    else if aere1 = "RELATED" then do; ord=2; label="  Related to AE"; output; end;
```

```
run;
```

```
data de_set;
```

```
    set de_1 de_2;
```

```
    output;
```

```
    if label = "Device Events" then do;
```

```
        ord=4;
```

```
        label="";
```

```
        %if &tab=1 %then %do;
```



```
        call missing (N_THSM2_2, N_SA, N_MCC, N_TOTAL, N_PRODUCT_TEST, E_THSM2_2,  
E_SA, E_MCC, E_TOTAL, E_PRODUCT_TEST);
```

```
    %end;
```

```
    %if &tab=2 %then %do;
```

```
        call missing(N_THSM2_2, E_THSM2_2);
```

```
    %end;
```

```
    output;
```

```
end;
```

```
run;
```

```
proc sort data=de_set; by ord; run;
```

```
proc sql noprint;
```

```
    *device events by severity - major/minor;
```

```
    create table sev as
```

```
    select trta, trtan, DESEV, count(distinct(usubjid))
```

```
    from de
```

```
    where DESEV ne "" and &where
```

```
    group by trta, trtan, DESEV;
```

```
    create table sev_ev as
```

```
    select trta, trtan, DESEV, count(usubjid)
```

```
    from de
```

```
    where DESEV ne "" and &where
```

```
    group by trta, trtan, DESEV;
```

*device events by severity - categories;

create table cat as

select trta, trtan, DESEV, DEDECOD, count(distinct(usubjid))

from de

where DESEV ne "" and &where

group by trta, trtan, DESEV, DEDECOD;

create table cat_ev as

select trta, trtan, DESEV, DEDECOD, count(usubjid)

from de

where DESEV ne "" and &where

group by trta, trtan, DESEV, DEDECOD;

quit;

*device events by severity - major/minor;

proc transpose data=sev out=t_sev prefix=N_;

id trta;

var _temg001;

by dese;

run;

proc transpose data=sev_ev out=t_sev_ev prefix=E_;

id trta;

var _temg001;

by dese;

```

run;

data sev_1;

length label $200;

    merge t_sev t_sev_ev;

    by _name_ desevel;

    if desevel = "MAJOR" then do; ord=5; num=1; label="  Major"; output; end;

    if desevel ne "MINOR" then do;

        desevel="MINOR";

        ord=6;

        num=2;

        label="  Minor";

        %if &tab=1 %then %do;

            call missing (N_THSM2_2, N_SA, N_MCC, N_TOTAL, E_THSM2_2, E_SA, E_MCC,
E_TOTAL);

        %end;

        %if &tab=2 %then %do;

            call missing(N_THSM2_2, E_THSM2_2);

        %end;

        output;

    end;

    else if desevel = "MINOR" then do; ord=6; num=2; label="  Minor"; output; end;

run;

```

```
*device events by severity - categories;
```

```
proc sort data=cat; by desevededecod; run;
```

```
proc transpose data=cat out=t_cat prefix=N_;
```

```
    id trta;
```

```
    var _temg001;
```

```
    by desevededecod;
```

```
run;
```

```
proc sort data=cat_ev; by desevededecod; run;
```

```
proc transpose data=cat_ev out=t_cat_ev prefix=E_;
```

```
    id trta;
```

```
    var _temg001;
```

```
    by desevededecod;
```

```
run;
```

```
data sev_2;
```

```
length label $200;
```

```
    merge t_cat t_cat_ev;
```

```
    by _name_ desevededecod;
```

```
    if missing(dedecod) then dedecod="Missing";
```

```
    label="  " || propcase(dedecod);
```

```
    if deseved="MAJOR" then do; num=1; ord=5; maj=_n_; end;
```

```
    if deseved="MINOR" then do; num=2; ord=6; min=_n_; end;
```

```
    line=_n_;
```

```
run;

%if &tab=1 %then %do;

    proc sort data=sev_2; by descending n_total; run;

%end;

%if &tab=2 %then %do;

    proc sort data=sev_2; by descending E_THSM2_2; run;

%end;
```

```
data sev_set;

    set sev_1 sev_2;

run;

proc sort data=sev_set; by ord; run;
```

```
data all;

    set de_set sev_set;

    output;

    if ord=4 then do;

        ord=4.5;

        label="Device events by severity";

        output;

    end;

run;
```

```
data all_1;

    set all;
```

```

%if &tab=1 %then %do;

if ord in (1,2,3,5,6) then do;

    if n_SA = . then n_SA = 0;

    if n_THSm2_2 = . then n_THSm2_2 = 0;

    if n_MCC = . then n_MCC = 0;

    if n_PRODUCT_TEST = . then n_PRODUCT_TEST = 0;

    if n_TOTAL = . then n_TOTAL = 0;

end;

%end;


%if &tab=2 %then %do;

drop n_TOTAL e_TOTAL;

if ord in (1,2,3,5,6) then do;

    if n_THSm2_2 = . then n_THSm2_2 = 0;

end;

%end;

run;


data rand_&pd (drop=_: n_THSm2_2 aere1 dedecod dese1 maj min);

    set all_1;


    tab=&tab;

    name="&name";

    pd=&pd;

```

```

*pre-randomization;

%if &tab=1 %then %do;

if ord in (1,2,3,5,6) then do;

    if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&B_SA)*100),.01),5.1)) || ")");

    if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&B_THS)*100),.01),5.1)) || ")");

    if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&B_MCC)*100),.01),5.1)) || ")");

    np_PRODT=strip(put(N_PRODUCT_TEST,8.) || " (" ||
compress(put(round(((N_PRODUCT_TEST/&B_ProdT)*100),.01),5.1)) || ")");

    np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&B_Total)*100),.01),5.1)) || ")");

end;

if ord in (1,2,3,5,6) then do;

    if e_SA = . then e_SA = 0;

    if e_THSm2_2 = . then e_THSm2_2 = 0;

    if e_MCC = . then e_MCC = 0;

    if e_PRODUCT_TEST = . then e_PRODUCT_TEST = 0;

    if e_TOTAL = . then e_TOTAL = 0;

end;

else do;

    call missing(e_SA, e_THSm2_2, e_MCC, e_PRODUCT_TEST, e_TOTAL);

end;

%end;

```

```

*after randomization;

%if &tab=2 %then %do;

    if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&A_THS)*100),.01),5.1)) || ")");

    if ord in (1,2,3,5,6) then do;

        if e_THSm2_2 = . then e_THSm2_2 = 0;

    end;

    else do;

        call missing(e_THSm2_2);

    end;

%end;

run;

%mend;

%pd (pd=1, tab=1, name=prior rand, where=%str(SAFBFL="Y" and ASPER=1));

%pd (pd=2, tab=2, name=after rand, where=%str(SAFAFL="Y" and ASPER in (2)));

%pd (pd=3, tab=2, name=after rand, where=%str(SAFAFL="Y" and ASPER in (3)));

%pd (pd=5, tab=2, name=after rand, where=%str(SAFAFL="Y" and ASPER in (2,3,4)));

data comb;

    set rand_1 rand_2 rand_3 rand_5;

    rec=_n_;

    output;

    *if ASPER=4 is missing;

```



```
if pd=3 and ord=6 then do;
    pd=4;
    ord=1;
    call missing(label, e_sa, e_product_test, e_thsm2_2, e_total, e_mcc, name, np_sa,
np_mcc, np_ths, np_total,
    np_prodt);
    output;
end;
run;
```

*may need to modify;

```
data dum1;
    pd=1; tab=1; label = " Related to AE"; ord=2; rec=1.5; output;

    pd=2; tab=2; label = " Related to AE"; ord=2; rec=8.5; output;

    pd=3; tab=2; label = " Related to AE"; ord=2; rec=21.5; output;

    pd=5; tab=2; label = " Related to AE"; ord=2; rec=39.5; output;
run;
```

```
proc sort data=comb; by tab pd rec ord label; run;
```

```
data comb_1;
    merge comb dum1;
    by tab pd rec ord label;
```

```

if pd=1 and label = " Related to AE" then do;

    e_sa=0; e_product_test=0; e_thsm2_2=0; e_mcc=0; e_total=0;

    np_sa="0 (0.0)"; np_mcc="0 (0.0)"; np_ths="0 (0.0)"; np_total="0 (0.0)"; np_prodt="0
(0.0)";

end;

if pd=2 and label = " Related to AE" then do;

    np_ths="0 (0.0)"; e_thsm2_2=0;

end;

if pd=3 and label = " Related to AE" then do;

    np_ths="0 (0.0)"; e_thsm2_2=0;

end;

if pd=5 and label = " Related to AE" then do;

    np_ths="0 (0.0)"; e_thsm2_2=0;

end;

run;

proc sort data=comb_1; by pd rec; run;

data final;

    set comb_1;

    by pd rec;

    blank="";

```

```
array a[5] NP_SA NP_THS NP_MCC NP_PRODT NP_TOTAL;  
array b[5] E_SA E_THSM2_2 E_MCC E_PRODUCT_TEST E_TOTAL;  
do i=1 to 5;  
    if a[i] = "0 (0.0)" then a[i] = "0";  
    if b[i] = 0 then b[i] = .;  
end;
```

```
if pd=1 then page=1;  
    *moving up confinement;  
    else if 43<=_n_<=50 then page=2;  
    else if 51<=_n_<=58 then page=3;  
    else if 59<=_n_<=63 then page=4;  
    else if 9<=_n_<=16 then page=5;  
    else if 17<=_n_<=22 then page=6;  
    else if 23<=_n_<=30 then page=7;  
    else if 31<=_n_<=38 then page=8;  
    else if 39<=_n_<=41 then page=9;  
    else if _n_=42 then page=10;  
run;
```

```
proc sort data=final; by page; run;
```

```
data final;  
    set final end=last;  
    by page;  
    call symput("page",compress(put(page,best.)));
```

```
run;
```

```
%put &page;
```

```
%let tflno = %str(T_15_02_06_15);
```

```
data tflds.&tflno;
```

```
    set final;
```

```
run;
```

```
/* Standard - leave this */
```

```
options number nodate orientation=landscape missing=' ';
```

```
%let linetop = \brdrt\brdrs\brdrw30; * needs to be 1.5pt so calculated in twips (1/20 pt) ;
```

```
%let linebot = \brdrb\brdrs\brdrw30;
```

```
/* Standard - macro for paging */
```

```
%macro outrtf(blankn=70, halfblnk=N);
```

```
%if &halfblnk=N %then %let halfblnk=;
```

```
%else %if &halfblnk=Y %then %let halfblnk=\~;
```

```
ods path stdlib.t106343 (read) ;
```

```
ods results off;
```

```
ods rtf toc_data file="/cvn/projects/prj/data/000000106343/TFL/&TFL_Part./Tables/&tflno..rtf"  
style=t106343 startpage=yes headery=1440 footery=1440 ;
```

```
ods noproctitle;
```

```
%do i=1 %to &page;
```

```
title ;
```

```
footnote;
```

```
%let wd=0;
```

```
%let NOOBS=0;
```

```
ods proclabel = ' ';
```

```
data comp;
```

```
    set final end=eof;
```

```
        where page=&i;
```

```
/*      comment it when data is available*/
```

```
    if pd=4 then call symput("noobs","1");
```

```
    %put  noobs=&noobs;
```

```
/* Amend title as needed */
```

```
        _firtitl="Table 15.2.6.15 Summary of THS 2.2 Menthol Device Events - Safety  
Population";
```

```
        _upcas=(length("Path: &TFLpath.")-  
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
```

```
        len=&blankn.-length("(Page &i of &page)");
```

```
        if eof then do;
```

```
            call symput('_FSRTITL', trim(left(_firtitl)));
```

```

        call symput('period1', compress(put(pd,best.)));

        call symput('rand1', compress(put(tab,best.)));

        call symput('_blankn', compress(put(len,best.)));

    end;

    drop _firtitl _upcas len;

run;

ods listing close;


* most set up in template others below;

* title arial 12pt bold with 12pt paragraph space below;

* all headers to be arial 11pt bold;

* data arial 10pt;

* headers to be central, text values left aligned and numeric centered around decimal point;

/* Update with your variables as needed */


proc report data = comp headline headskip nowd split = '$' %if &i=1 %then %do; contents=' ' %end;
%else %do; contents="" %end;;;

    column page label blank

%if &rand1.=1 %then %do; ("THSm2.2$(N=%left(&B_THS))&linebot" np_ths e_thsm2_2)
("mCC$(N=%left(&B_mCC))&linebot" np_mcc e_mcc)

("SA$(N=%left(&B_SA))&linebot" np_sa e_sa) ("Product Test$(N=%left(&B_Prodt))&linebot" np_prodt
e_product_test)

("Overall Safety$(N=%left(&B_Total))&linebot" np_total e_total) %end;

%if &rand1.=2 and &period1. ne 4 %then %do; ("THSm2.2$(N=%left(&A_THS))&linebot" np_ths
e_thsm2_2) %end;;

```

```

define page      / order order = internal noprint;

define label      /"Device Events" display style(column)={just=left cellwidth=4.2cm
asis=on} style(header)={just=left};

define blank      /"" display style(column)=[just=left cellwidth=1.0cm];

%if &rand1.=1 %then %do;

define np_ths      /"n (%)" display style={just=c cellwidth=1.4cm}
style(header)={just=center} ;

define e_thsm2_2    /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

define np_mcc      /"n (%)" display style={just=c cellwidth=1.2cm}
style(header)={just=center} ;

define e_mcc      /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

define np_sa      /"n (%)" display style={just=c cellwidth=1.2cm}
style(header)={just=center} ;

define e_sa      /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

define np_prodt    /"n (%)" display style={just=c cellwidth=1.4cm}
style(header)={just=center} ;

define e_product_test /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

define np_total    /"n (%)" display style={just=c cellwidth=1.1cm}
style(header)={just=center} ;

define e_total    /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

%end;

%if &rand1.=2 and &period1. ne 4 %then %do;

define np_ths      /"n (%)" display style={just=c cellwidth=1.4cm}
style(header)={just=center} ;

define e_thsm2_2    /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

```

%end;

break after page / page;

COMPUTE AFTER page;

line " ";

%IF &NOOBS. = 1 %THEN %DO;

LINE "No THS 2.2 events were reported in follow-up safety time period";

LINE " ";

%END;

ENDCOMP;

compute before page / style={protectspecialchars=off};

line "&linetop";

endcomp;

compute before _page_ / style={just=left protectspecialchars=off};

line "\b\fs24\sa24&_FSRTITL." ; * \b = bold, \fs24 is font size 12pt, \sa24 is space after 12pt;

line "";

%if &period1.=1 %then %do; line "Safety Time Period: Pre-Randomization Period" %end;

%if &period1.=2 %then %do; line "Safety Time Period: Confinement Period" %end;

%if &period1.=3 %then %do; line "Safety Time Period: Ambulatory Period" %end;

%if &period1.=4 %then %do; line "Safety Time Period: Safety Follow-up Period" %end;

%if &period1.=5 %then %do; line "Safety Time Period: Post-Randomization Period"

%end;


```

line "";

line "&linebot";

endcomp;

compute after _page_/ style={just=left protectspecialchars=off pretext="&linetop."};

%if &rand1.=1 %then %do;

line 'Note: "Product Test" refers to all subjects who tested the THS product but were not
randomized. The Overall Safety refers to all subjects exposed to THSm2.2.';

line 'Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THSm2.2 =
Tobacco Heating System 2.2 Menthol.';

line 'Note: Percentages are based on the number of subjects indicated in the column
header (N).';

line "";

%end;

%if &rand1.=2 and &period1. ne 4 %then %do;

line 'Note: THSm2.2 = Tobacco Heating System 2.2 Menthol.';

line 'Note: Percentages are based on the number of subjects indicated in the column
header (N).';

line "";

%end;

line 'Appendix 15.3.6.5';

line "Study ID: ZRHM-REXA-08-US   Program: &TFLprg   Status: &status"
&_blankn.*"\~" "&sysdate" &_blankn.*"\~" "(Page &i of &page)";

endcomp;

run;

```

```
%end;
```

```
ods rtf close;
```

```
ods results on;
```

```
ods path sashelp.tmplmst (read);
```

```
%mend ;
```

```
%outtrtf(blankn=50, halfblnk=N);
```

```
ods listing;
```

```
*=====;
```

```
* END OF PROGRAM CODE ;
```

```
*=====;
```

```
%m_logchk(showlog=yes);
```