

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106326;
%put NOTE: Client Protocol ID   : ZRHM-PK-05-JP;
%put NOTE: Program Name        : t_condis.sas;
%put NOTE: Purpose              : table of comcomitant diseases;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADMH ADAM.ADSL;
%put NOTE: Output               : t_15_2_1_7(cd);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jriley;
%put NOTE: Creation Date        : 2014-08-05;
%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: 11Aug2014   JR        1) Amended prog header;
%put NOTE: 11Aug2014   JR        2) Amended headers;
%put NOTE: 11Aug2014   JMH       3) Amended column header;
%put NOTE: ;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

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

/* Standard - just change the number to match the listing you're working
on. Also change the letters in the*/
/* bracket, eg ccb = current cigarette brands. Make sure to do this at
the top of the code too. */

      %let tflno=T_15_02_01_07(cd);

/* Standard - leave this */
%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", ""));
run;

*****;
* read in data ;
*****;

data adsl1;
    set adam.adsl;
    where saffl = 'Y';
    if missing(trtseqa) then delete;
    if index(trtseqa, 'Expos') then delete;
    headorder1=trtseqa;
    headtext1=trtseqa;
    output;
    trtseqa=99;
    headorder1=99;
    trtseqa='Overall Safety';
    headtext1='Overall Safety';
    output;
run;

data dumtrts; /*Use this to output any columns for which N=0*/
    attrib headtext1 length=$200.
            headorder1 length=8.;
    headorder1=1;
    headtext1='THS 2.2 Menthol - mCC';
    output;
    headorder1=2;
    headtext1='mCC - THS 2.2 Menthol';
    output;
    headorder1=3;
    headtext1='THS 2.2 Menthol - NRT gum';
    output;
    headorder1=4;
    headtext1='NRT gum - THS 2.2 Menthol';
    output;
    headorder1=5;
    headtext1='Enrolled not randomized';
    output;
run;

proc sort data=adsl1; by headorder1 headtext1; run;

proc freq data=adsl1 noprint;
    table headorder1*headtext1/ out =tot(drop=percent);
run;

```

```

data adsl;
    merge tot(in=a) dumtrts(in=b);
    by headorder1 headtext1;
    if a or b;
    if b and not a then count=0;
run;

data tot2;
    set adsl;
    call symput('trt' || compress(put(headorder1,best.)),
compress(count));
run;
%macro test;

%if %sysfunc(exist(adam.admh))=0 %then %do;
data paging;
    page=1; flag=1; ln=1; sort2=.; sort3=.; sort4=.; odd=.; odd2=.;
column=''; n1=''; p1=''; e1=''; n2=''; p2=''; e2='';
    n3=''; p3=''; e3=''; n5=''; p5=''; e5=''; n99=''; p99=''; e99=''; n4='';
p4=''; e4=''; mhbodsys='';
    output;
    call symput("page",'1');

        attrib n1 label = " n"
                n2 label = " n"
                n3 label = " n"
                n4 label = " n"
                n5 label = " n"
                n99 label = " n"
                p1 label = ' (%) '
                p2 label = ' (%) '
                p3 label = ' (%) '
                p4 label = ' (%) '
                p5 label = ' (%) '
                p99 label = ' (%) '
                e1 label = "Events"
                e2 label = "Events"
                e3 label = "Events"
                e4 label = "Events"
                e5 label = "Events"
                e99 label = "Events";

run;
%end;

%else %if %sysfunc(exist(adam.admh)) %then %do;
data mh1;
    set adam.admh;
    where anycdf1='Y' and saffl='Y' and mhcat='CONCOMITANT DISEASE';
/*Only keep subjects with Concomitant Diseases*/
    if missing(trtseqan) then delete;
    if index(trtseqa,'Expos') then delete;
run;

data mh;

```

```

        set mh1;
        headorder1=trtsega;
        headtext1=trtsega;
        output;
        headorder1=99;
        headtext1='Overall Safety';
        output;
run;

proc sort data=mh; by headorder1 headtext1 usubjid; run;

proc sort data=adsl1(keep=usubjid headorder1 headtext1); by headorder1
headtext1 usubjid; run;

data mh02;
    merge mh(in=a) adsl1;
    by headorder1 headtext1 usubjid;
    if a;
run;

proc sort data=mh02; by headorder1 headtext1; run;

* Create an additional observation with missing VOL value for each table
section;
* This is used to ensure that all table rows are output, even for rows
with no adverse events;

data mh03;
    set mh02;
    output;
    mhbodsys='Any concomitant disease';
    output;
run;

/** Number of Ae's overall **/
proc freq data=mh03 noprint;
    tables headorder1*headtext1*mhbodsys / out=ovall(rename=(count=tot)
drop=percent);
run;

/** getting number of subjects studied ***/
proc sort data=mh03 out=nmh3 nodupkey;
    by headorder1 headtext1 mhbodsys subjidn;
run;

proc freq data=nmh3 noprint;
    tables headorder1*headtext1*mhbodsys / out=novall(rename=(count=ntot)
drop=percent);
run;

data otot;
    merge ovall novall;
    by headorder1 headtext1 mhbodsys;
run;

```

```

    /*** number of subjects and aes in overall **/
    data overall;
        set otot;
            by headorder1 headtext1 mhbodsys;
            if mhbodsys='Any concomitant disease' then sort2=1;
            else sort2=2;
        sort3=0;
    run;

    /*** bodsys ordered ****/
    proc freq data=mh03 noprint;
        tables headorder1*headtext1*mhbodsys / out=tmhbod(rename=(count=tot)
drop=percent);
    run;

    /*** getting number of subjects studied ****/
    proc sort data=mh03 out=nmh2 nodupkey;
        by headorder1 headtext1 mhbodsys subjidn;
    run;

    proc freq data=nmh2 noprint;
        tables headorder1*headtext1*mhbodsys / out=mhbod(rename=(count=ntot)
drop=percent);
    run;

    data body;
        merge tmhbod mhbod;
        by headorder1 headtext1 mhbodsys;
    run;

    data body2;
        set body;
            if mhbodsys='Any concomitant disease' then do; sort2=1;
sort3=0; end;
            else do; sort2=2; sort3=0; end;
    run;

    /** bodysystem by prefterm **/
    proc freq data=mh03 noprint;
        tables headorder1*headtext1*mhbodsys*mhdecod /
out=preft(rename=(count=tot) drop=percent);
    run;

    /*** getting number of subjects studied ****/
    proc sort data=mh03 out=npmh2 nodupkey;
        by headorder1 headtext1 mhbodsys mhdecod subjidn;
    run;

    proc freq data=npmh2 noprint;
        tables headorder1*headtext1*mhbodsys*mhdecod /
out=npreftr(rename=(count=ntot) drop=percent);
    run;

```

```

data prefterm;
  merge preft npreft;
  by headorder1 headtext1 mhbodsys mhdecod;
run;

data mhdecod;
  set prefterm;
  if mhbodsys='Any concomitant disease' then do; sort2=1;
sort3=0; end;
  else do; sort2=2; sort3=1; end;
run;

data all;
  set overall body2 mhdecod;
run;

proc sort data=all out=all2;
  by headorder1 headtext1 sort2 mhbodsys mhdecod;
run;

data format;
  merge all2(in=a) dumtrts tot;
  by headorder1 headtext1;
  if not a then do;
    sort2=1;
    sort3=0;
    mhbodsys='Any concomitant disease';
  end;
run;

data format2;
  set format;
  attrib text text2 text3 format=$20.;
  /* Percentage of subjects*/
  if not missing(count) then percent=put((ntot/count)*100,8.1);
  else percent='0';

  /*n value*/
  if missing(ntot) then text='0';
  else text=put(ntot,3.);

  /*% value*/
  if not missing(percent) and percent ne 0 then do;
    if percent=100 then text3=' (100 %)';
    else if percent ge 10 then text3=' (
'||compress(put(percent,8.1))||'%)';
    else if percent lt 10 then text3=' (
'||compress(put(percent,8.1))||'%)';

    if index(text3,'100') then
text3=tranwrd(text3,'100.0','100');
  end;

```

```

        /*events value*/
        if missing(tot) then text2='';
    else text2=compress(put(tot,3.));

        if mhbodsys='Any concomitant disease' and not
missing(mhdecod) then delete;
        drop percent;
    run;

    proc sort data=format2 nodupkey; by headorder1 headtext1 mhbodsys tot
ntot sort2 sort3 mhdecod count text text2 text3; run;

    proc sort data=format2; by headorder1 headtext1 sort2 mhbodsys sort3
mhdecod; run;

    proc sort data=format2 out=format3; by sort2 mhbodsys sort3 mhdecod;
run;

    proc transpose data=format3 out=nformat prefix=n;
        by sort2 mhbodsys sort3 mhdecod;
        var text;
        id headorder1;
        idlabel headtext1;
    run;

    proc transpose data=format3 out=eformat prefix=e;
        by sort2 mhbodsys sort3 mhdecod;
        var text2;
        id headorder1;
        idlabel headtext1;
    run;

    proc transpose data=format3 out=pformat prefix=p;
        by sort2 mhbodsys sort3 mhdecod;
        var text3;
        id headorder1;
        idlabel headtext1;
    run;

    data tformat;
        merge nformat eformat pformat;
        by sort2 mhbodsys sort3 mhdecod;
    run;

    data sformat3;
        set format3;
        col=headorder1;
    run;

    proc sort data=sformat3;

```

```

    by sort2 mhbodsys sort3 mhdecod;
run;

proc transpose data=sformat3(where=(sort3=0)) out=tsort prefix=n;
    by sort2 mhbodsys sort3 mhdecod;
    var ntot;
    id col;
run;

proc transpose data=sformat3(where=(sort3=0)) out=tsort_a prefix=t;
    by sort2 mhbodsys sort3 mhdecod;
    var tot;
    id col;
run;

data tsort1;
    merge tsort tsort_a;
    by sort2 mhbodsys sort3 mhdecod;
run;

data tsort2;
    set tsort1;
    num=t99;
run;

proc sort data=tsort2;
    by sort2 sort3 descending num;
run;

/** unique sorting numbers for bodsystems by total number of aes**/
data sorting;
    set tsort2;
    by sort2 sort3 descending num;
    if first.sort3 then odd=1;
    else odd+1;
    keep sort2 mhbodsys odd;
run;

proc sort data=sorting;
    by sort2 mhbodsys;
run;

data final;
    merge tformat sorting;
    by sort2 mhbodsys;
run;

proc transpose data=sformat3(where=(sort3=1)) out=psort prefix=n;
    by sort2 mhbodsys sort3 mhdecod;
    var ntot;
    id col;
run;

proc transpose data=sformat3(where=(sort3=1)) out=psort_a prefix=t;

```



```

    by sort2 mhbodsys sort3 mhdecod;
    var tot;
    id col;
run;

data psort1;
    merge psort psort_a;
    by sort2 mhbodsys sort3 mhdecod;
run;

data psort2;
    set psort1;
    num=0;
    run;

proc sort data=psort2;
    by sort2 sort3 descending num;
run;

/** unique sorting numbers for preterm by total number of aes**/
data psorting;
    set psort2;
    by sort2 sort3 descending num;
    if first.sort3 then odd2=1;
    else odd2+1;
    keep sort2 mhbodsys mhdecod odd2;
run;

proc sort data=psorting;
    by sort2 mhbodsys mhdecod;
run;

proc sort data=final;
    by sort2 mhbodsys mhdecod;
run;

data final2;
    merge final psorting;
    by sort2 mhbodsys mhdecod;
run;

proc sort data=final2;
    by sort2 odd mhbodsys odd2 mhdecod;
run;

data final3;
    set final2;
    attrib column column1 wrap format=$400.;

    wrap = mhdecod;

if sort3=1 then do;
i=35; *This is the max length allowed on a single line - change as
needed;

```

```

if length(wrap)>i then do;
  nwraps = int(length(wrap)/i); *Calculate how many lines the text will
  wrap over;
  do while(nwraps > 0);
    fin=0;
    j = i*nwraps; *Calculate starting point - loop will cycle backwards
    from this point looking for a space;
    do while(fin=0 AND J GT 1);
      if substr(wrap,j,1)=' ' then do;
        wrap=substr(wrap,1,j-1) || "|n |S={foreground=white} . |S={} " ||
substr(wrap,j+1);
        fin=1;
      end;
      else j=j-1; *No space found - move back one character;
    end;
    nwraps=nwraps-1; *Once this wrap is handled, move up a line until all
    are handled (when nwraps = 0);
  end;
end;

column="|S={foreground=white} . |S={} " || wrap;
end;

```

```

    if sort2=2 and sort3=0 then column=trim(mhbodsys);
    if sort2=1 then do; column=upcase(mhbodsys); end;

    if missing(sort2) and missing(sort3) then delete;

        column1=left(trim(tranwrd(column,'|S={foreground=white} .
|S={} ','')));
        column1=left(trim(tranwrd(column1,'|n','')));

    run;

```

```

proc sort data=final3(where=(sort3=0)) out=blanks;
  by sort2 sort3;
run;

```

```

data blanks1;
  set blanks;
      if (n1=0 or missing(n1)) and (e1=0 or missing(e1)) then
del1=1;
      if (n2=0 or missing(n2)) and (e2=0 or missing(e2)) then
del2=1;
      if (n3=0 or missing(n3)) and (e3=0 or missing(e3)) then
del3=1;
      if (n4=0 or missing(n4)) and (e4=0 or missing(e4)) then
del4=1;
      if (n5=0 or missing(n5)) and (e5=0 or missing(e5)) then
del5=1;
  keep sort2 sort3 mhbodsys del;;
run;

```

```

proc sort data=final3; by sort2 mhbodsys; run;
proc sort data=blanks1; by sort2 mhbodsys; run;

data final3a;
    merge final3 blanks1;
    by sort2 mhbodsys;
run;

proc sort data=final3a;
    by sort2 odd mhbodsys sort3 odd2 column;
run;

data labels;
    set final3a;
        attrib n1 label = "n"
                n2 label = "n"
                n3 label = "n"
                n4 label = "n"
                n5 label = "n"
                n99 label = "n"
                p1 label = "(%) "
                p2 label = "(%) "
                p3 label = "(%) "
                p4 label = "(%) "
                p5 label = "(%) "
                p99 label = "(%) "
                e1 label = "Events"
                e2 label = "Events"
                e3 label = "Events"
                e4 label = "Events"
                e5 label = "Events"
                e99 label = "Events"
                column label = "MH text";

    if sort3=0 then do;
        del1=.;
        del2=.;
        del3=.;
        del4=.;
        del5=.;
    end;
run;

data final4;
    set labels;

    array a [4] n1 n2 n4 n5; /*n3 and e3 are left missing as there are
no comcomitant diseases in that category*/
    array b [4] e1 e2 e4 e5;
    array c [4] del1 del2 del4 del5;
do i=1 to 4;
    if c[i] ne 1 then do;
        if missing(a[i]) then a[i] ='0';

```

```

end;

end;

if missing(n99) then n99='0';

flag=1;
if not index(column,'|S={') then do;

column2=uppercase(substr(column,1,1))||lowercase(substr(column,2));
end;
else column2=column;
drop column;
rename column2=column;
run;

proc sql noprint;

create table table.t_15_02_01_07 as
select mhbodsys, mhdecod, column, column1, n1, n2, n3, n4, n5, n99,
e1, e2, e3, e4, e5, e99, p1, p2, p3, p4, p5, p99
from final4
order by sort2, odd, mhbodsys, sort3, odd2, column;

quit;

data paging;
set final4;
by sort2 odd mhbodsys sort3 odd2 column;
if (ln gt 8 and first.mhbodsys) then ln=1;
else ln+1;

if ln=1 then page+1;
call symput("page",compress(put(page,best.)));

retain sort4;
if first.mhbodsys then sort4+1;

run;
%end;
%mend;
%test;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&p_pgsz missing='
';
ods escapechar='|';
%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=, halfblnk=);

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

```

```

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

ods path stdlib.tl06326 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106326/TFL/&TFL_Part./&tflno..rtf"
style=tl06326 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

title ;
footnote;
%let wd=0;
%let noobs=0;

%let maxpage=%eval(&page);

ods proclabel = ' ';

data comp;
    set paging end=eof;
    where page=&i;
if column='' and sort3=. then call symput('noobs', "1");
    /* Amend title as needed */
        _firtitl="Table 15.2.1.7 Summary of Concomitant Diseases -
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('_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 missing nowd split = '$' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents=' ' %end;;
    column flag page sort2 odd sort4 sort3 mhbodsys odd2 ("System
Organ Class" ("Preferred Term" column)) /* 3) JMH 11Aug2014 */

```

```

("Sequence &linebot" ("THS 2.2 Menthol -$mCC $(N=&trt1) &linebot" ("n (%)
Events" n1 p1 e1)) /* 2) JR 11Aug2014 */ /* 3) JMH 11Aug2014 */
("mCC -$THS 2.2 Menthol$(N=&trt2) &linebot" ("n (%) Events" n2 p2 e2))
/* 2) JR 11Aug2014 */ /* 3) JMH 11Aug2014 */
("THS 2.2 Menthol -$NRT gum $(N=&trt3) &linebot" ("n (%) Events" n3 p3
e3)) /* 2) JR 11Aug2014 */
("NRT gum -$THS 2.2 Menthol$(N=&trt4) &linebot" ("n (%) Events" n4 p4
e4)) /* 2) JR 11Aug2014 */
("Enrolled Not$Randomized$(N=&trt5) &linebot" ("n (%) Events" n5 p5 e5)))
("Overall$Safety$(N=&trt99) &linebot" ("n (%) Events" n99 p99 e99));
    define flag      / order order = internal noprint;
    define page      / order order = internal noprint;
    define flag      / order order = internal noprint;
    define page      / order order = internal noprint;
    define sort2     / order order=internal noprint;
define odd          / order order=internal noprint;
define sort4        / order order=internal noprint;
define sort3        / order order=internal noprint;
define mhbodsys     / order order=internal noprint;
define odd2         / order order=internal noprint;

    define column    / group style={just=left cellwidth=4cm}
/*"|\~ {Preferred Term}"*/ "" style(header)={just=center}; /* 3) JMH
11Aug2014 */
    define n1        / display style={just=d cellwidth=0.2cm}
style(header)={just=center}"";
    define n2        / display style={just=d cellwidth=0.2cm}
style(header)={just=center}"";
    define n3        / display style={just=d cellwidth=0.2cm}
style(header)={just=center}"";
    define p1        / display style={just=d cellwidth=1cm}
style(header)={just=center}"";
    define p2        / display style={just=d cellwidth=1cm}
style(header)={just=center}"";
    define p3        / display style={just=d cellwidth=1cm}
style(header)={just=center}"";
    define e1        / display style={JUST=L cellwidth=0.5CM}
style(header)={just=center}"";
    define e2        / display style={JUST=L cellwidth=0.5CM}
style(header)={just=center}"";
    define e3        / display style={JUST=L cellwidth=0.5CM}
style(header)={just=center}"";
    define n4        / display style={just=d cellwidth=0.2cm}
style(header)={just=center}"";
    define n5        / display style={just=d cellwidth=0.2cm}
style(header)={just=center}"";
    define n99       / display style={just=d cellwidth=0.2cm}
style(header)={just=center}"";
    define p4        / display style={just=d cellwidth=1cm}
style(header)={just=center}"";
    define p5        / display style={just=d cellwidth=1cm}
style(header)={just=center}"";
    define p99       / display style={just=d cellwidth=1cm}
style(header)={just=center}"";

```

```

        define e4          / display style={JUST=L cellwidth=0.5CM }
style(header)={just=center}"";
        define e5          / display style={JUST=L cellwidth=0.5CM }
style(header)={just=center}"";
        define e99        / display style={JUST=L cellwidth=0.5CM }
style(header)={just=center}"";

break before flag / page %if &i=1 %then %do;
contents("&_fsrtitl" %end; %else %do; contents='' %end;;

break after page / page;

compute before page / style={protectspecialchars=off};;
    line "&linetop";
endcomp;

compute after page/style={just=center cellwidth=5cm
protectspecialchars=off};;
    %if &NOOBS.=1 %then %do;
    line "No concomitant disease data recorded";
    line " ";
    %end;
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 "&linebot";
endcomp;

compute after _page_ / style={just=left protectspecialchars=off
PRETEXT="&LINETOP."};;
    line 'Note: mCC = menthol conventional cigarettes; NRT gum =
Nicotine Replacement Therapy gum; THS = Tobacco Heating System.';
    line 'Note: Enrolled Not Randomized refers to all subjects
enrolled but not randomized. Overall Safety refers to all subjects
exposed to THS 2.2 Menthol or NRT gum.';
    LINE 'Note: Percentages are based on the number of subjects
indicated in the column header (N).';
    line ' ';
    LINE 'Appendix 15.3.1.8.2';
    line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of
&maxpage)";
    line "Program Run: &sysdate &sysuserid Program Status:
&status";
endcomp;
run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

```

```

%mend ;

%outrtf(blankn=70, halfblnk=N);
%macro check;
%if %sysfunc(exist(adam.admh)) %then %do;
ods listing;
proc printto print = "&table./t_15_02_01_07.lst" new;
run;

proc contents data = table.t_15_02_01_07 varnum;
run;
ods listing close;
%end;
%mend;
%check;
proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```