

```

%_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_pdev.sas;
%put NOTE: Purpose              : table of protocol deviations;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADDV ADAM.ADSL;
%put NOTE: Output               : T_15_2_1_3(pdev);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jriley;
%put NOTE: Creation Date        : 2014-08-06;
%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 headers;
%put NOTE: 23Sep2014   JR        2) Amended format as per client
comments/ REX studies;
%put NOTE: 23Sep2014   JR        3) Amended zero presentation;
%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_03(pdev);

/* 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 dumtrts1; /*Use this to output any columns for which N=0*/
    attrib trtsega length =$200.
           trtseqan length=8.;

    trtseqan=6;
    trtsega='Exposed not randomized';
    output;
run;

data adsl;
    set adam.adsl;
    where saffl = 'Y' and enrfl = 'Y';
    if trtseqan=5 then do;
        trtseqan=6;
        trtsega='Exposed not randomized';
    END;
    output;
    trtseqan=99;
    trtsega='Overall Safety';
    output;
run;

data addv;
    set adam.addv;
    where saffl = 'Y' and enrfl = 'Y';
    if trtseqan=5 then do;
        trtseqan=6;
        trtsega='Exposed not randomized';
    end;
    output;
    trtseqan=99;
    trtsega='Overall Safety';
    output;
run;

proc freq data=adsl noprint;
    table trtseqan*trtsega/ out =tota(drop=percent);
run;

data tot;
    merge tota(in=a) dumtrts1(in=b);
    by trtseqan trtsega;

```

```

        if a or b;
            if b and not a then do;
                count = 0;
            end;
            headorder1=trtseqan;
            headtext1=trtsega;
            rename count=total;
run;

data tot2;
    set tot;
    call symput('trt' || compress(put(trtseqan,best.)),
compress(total));
run;

data deviation02;
    set addv;

    headorder1=trtseqan;
    headtext1=trtsega;
run;

proc sort data=deviation02;
    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 device events;
data deviation03;
    set deviation02;
    by headorder1 headtext1;
    dvdecod=avalc;
run;

* Create values for table rows;
data deviation04;
    set deviation03;
    length rowtext $200;
    * Major deviations;
    roworder1 = 2;
    roworder2 = 1;
    rowtext = 'Major';
    if dvsig='MAJOR' then output;
    * Minor deviations;
    roworder1 = 3;
    roworder2 = 1;
    rowtext = 'Minor';
    if dvsig='MINOR' then output;
run;

```

```

data addv1;
    set addv;
    headorder1=trtseqan;
    headtext1=trtsega;

    dvdecod=avalc;
    drop trtseqan trtsega;
run;

proc sql;
    create table results01 as
    select headorder1, headtext1, count(distinct usubjid) as treated
    from addv1
    group by headorder1, headtext1;
quit;

proc sort data=deviation04 out=deviation04_a nodupkey dupout=dups;
    by headorder1 headtext1 roworder1 roworder2 rowtext usubjid dvcat
param dvdecod dvsig avisit atpt;
run;

proc sql;
    create table results02 as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
    dvcat,param, dvdecod, count(dvdecod) as events,
        count(distinct subjid) as subjects
    from deviation04_a
    group by headorder1, headtext1, roworder1, roworder2, rowtext, dvcat,
param;
quit;

/* START 2) JR 23Sep2014 */
PROC SQL;
    CREATE TABLE RESULTS02_SIG AS
    SELECT HEADORDER1, HEADTEXT1, ROWORDER1, ROWORDER2, ROWTEXT,
COUNT(ROWTEXT) AS EVENTS,
        COUNT(DISTINCT SUBJID) AS SUBJECTS
    FROM DEVIATION04_A
    GROUP BY HEADORDER1, HEADTEXT1, ROWORDER1, ROWORDER2, ROWTEXT;
QUIT;

DATA SIGS;
    SET RESULTS02_SIG;
    ROWORDER2=0;
RUN;

DATA RESULTS02_EXTRA;
    SET RESULTS02_SIGS;
    BY HEADORDER1 HEADTEXT1;
RUN;
/* END 2) JR 23Sep2014 */

data results03;

```

```

merge results02_EXTRA(in=a) results01(keep=headorder1 headtext1
treated); /* 2) JR 23Sep2014 */
  by headorder1 headtext1;
  if a;
run;

data results04;
  set results03;
run;

proc sort data=results04;
  by headorder1 headtext1 roworder1 roworder2 rowtext;
run;

* Create data set with all combinations of row values and column values;
* This creates a data set with an observation for each table cell;
proc sql;
  create table results05 as
  select *
  from (select distinct headorder1, headtext1, roworder1, roworder2,
rowtext from results04);
quit;

* Sort the all combinations data set by section heading order, row order
and column order;
proc sort data=results05;
  by headorder1 headtext1 roworder1 roworder2 rowtext ;
run;

* Merge the results data set with the all combinations data set;
* This effectively adds observations with missing results for table cells
with no results;
* This allows text to be created for these table cells if necessary;
data results06;
  merge results04 results05;
  by headorder1 headtext1 roworder1 roworder2 rowtext ;
run;

* Convert results to text values for the summary table;
data results07;
  merge results06 tot;
  by headorder1 headtext1;
  length text text2 text3 $20. ;
  if (events = . and subjects = .) or missing(events) and
missing(subjects) then do;
    events = 0;
    subjects = 0;
  end;
  percent = 100 * subjects / TOTAL;

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

```

```

        /*% value*/
        if missing(percent) then text3='';
        else 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))||'%)';

        /*events value*/
        if missing(events) then text2='0';
        else text2=compress(put(events,3.));

    keep headorder1 headtext1 roworder1 roworder2 rowtext dvcat param text
    text2 text3;
run;

proc sort data=results07 nodupkey;
    by headorder1 headtext1 roworder1 roworder2 rowtext dvcat param
    text text2 text3;
run;

data dumtrts_MAJ; /*Use this to output any columns for which N=0*/
    attrib headtext1 length=$200.
            rowtext length=$70.
            headorder1 length=8.;

    roworder1=2;
    roworder2=0/*1*/; /* 2) JR 23Sep2014 */
    rowtext='Major';
    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=6;
    headtext1='Exposed not randomized';
    output;
    headorder1=99;
    headtext1='Overall Safety';
    output;
run;

data dumtrts_min;
    attrib headtext1 length=$200.
            rowtext length=$70.
            headorder1 length=8.;

```

```

roworder1=/*2*/3; /* 2) JR 23Sep2014 */
roworder2=/*1*/0; /* 2) JR 23Sep2014 */
rowtext='Minor';
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=6;
headtext1='Exposed not randomized';
output;
headorder1=99;
headtext1='Overall Safety';
output;
run;

data dumtrts;
    set dumtrts_maj dumtrts_min;
run;

proc sort data = dumtrts;
    by headorder1 headtext1 roworder1 roworder2 rowtext;
run;

data results07a;
    merge results07(in=a) dumtrts(in=b);
    by headorder1 headtext1 roworder1 roworder2 rowtext;
    if a or b;
    IF B AND NOT A THEN DO; /* start 2) JR 23Sep2014 */
    TEXT='0';
    TEXT2='0';
    END; /* end 2) JR 23Sep2014 */
run;

proc sort data=results07a;
    by roworder1 roworder2 rowtext dvcat param;
run;

* Transpose the results;
proc transpose data=results07a out=results08_n prefix=n ;
    by roworder1 roworder2 rowtext dvcat param;
    id headorder1;
    idlabel headtext1;
    var text ;
run;

proc transpose data=results07a out=results08_p prefix=p ;

```

```

    by roworder1 roworder2 rowtext dvcat param;
    id headorder1;
        idlabel headtext1;
    var text3;
run;

proc transpose data=results07a out=results08_e prefix=e ;
    by roworder1 roworder2 rowtext dvcat param;
    id headorder1;
        idlabel headtext1;
    var text2 ;
run;

data results08;
    merge results08_n results08_e results08_p;
    by roworder1 roworder2 rowtext dvcat param;

/*    if missing(dvcat) then delete;    */
    IF ROWORDER2=0 THEN PARAM=ROWTEXT; /* 2) JR 23Sep2014 */

    n99num=input(n99,best.);
run;

proc sort data=results08;
    by roworder1 roworder2 descending n99;
run;

data results08_x;
    set results08;
    by roworder1 roworder2 descending n99;
    sortord+1;
run;

data labels;
    set results08_x;
    attrib
        N: LABEL ="n"
        P: LABEL = "(%) "
        E: LABEL = "Events"
        rowtext label = "Deviation Class";

    if p1 = '( 0.0%)' then p1 = '';
    if p2 = '( 0.0%)' then p2 = '';
    if p3 = '( 0.0%)' then p3 = '';
    if p4 = '( 0.0%)' then p4 = '';
    if P6 = '( 0.0%)' then P6 = '';
    if p99 = '( 0.0%)' then p99 = '';

    attrib wrap length = $200
        PARAM1 LENGTH=$200.; /* 2) JR 23Sep2014 */
    wrap = /*rowtext*/PARAM; /* 2) JR 23Sep2014 */

```



```

/*This isn't needed for this table so will change roworder value so it
doesn't change our output*/
if /*roworder2 ne 1 and*/ roworder2 ne /*2*/0 then do; /* 2) JR
23Sep2014 */
    i=22; *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);
                if substr(wrap,j,1)=' ' then do;
                    /*rowtext*/PARAM1=substr(wrap,1,j-1) || "$n
$S={foreground=white} . $S={} " || substr(wrap,j+1); /* 2) JR 23Sep2014
*/
                    fin=1;
                end;
                else j=j-1; *no space found - move back one character;
            end;
            nwraps=nwraps-1; *once this wrap is handled, move up a line
until all are handled (when nwraps = 0);
        end;
        /*rowtext*/PARAM1='$S={foreground=white} . $S={} ' ||
left(trim(/*rowtext1*/PARAM1)); /* 2) JR 23Sep2014 */
    end;
    else do;
        /*rowtext*/PARAM1='$S={foreground=white} . $S={} ' ||
left(trim(/*rowtext1*/PARAM1)); /* 2) JR 23Sep2014 */
    end;

end;
else do; /*rowtext1*/PARAM1=/*rowtext*/PARAM; end; /* 2) JR 23Sep2014 */

/* 3) JR 23Sep2014 */
/* array a [4] n1 n2 n3 n4;*/
/* array b [4] e1 e2 e3 e4;*/
/* do i=1 to 4;*/
/* if missing(a[i]) then a[i] ='0';*/
/* if missing(b[i]) then b[i] ='0';*/
/* end;*/

        if missing(n99) then n99='0';
        if missing(e99) then e99='0';

        if missing(n6) then n6='0';

    flag = 1;
run;

proc sql noprint;

```

```

create table table.t_15_02_01_03 as
select /*rowtext*/PARAM, /*rowtext1*/PARAM1, dvcat, n1, n2, n3, n4, n6,
n99, p1, p2, p3, p4, p6, p99 /* 3) JR 23Sep2014 */
from labels
order by roworder1, roworder2, sortord;

quit;

/*options noreplace;*/

data paging;
  set labels;
    by roworder1 roworder2 sortord dvcat param;
      if first.roworder1 AND LN > 5 then ln=1; /*Amend to look
presentable, and avoid page overflows*/ /* 2) JR 23Sep2014 */
      else ln+1;
      if ln=1 then page+1;
      call symput("page",compress(put(page,best.)));
run;

/* 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.t106326 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106326/TFL/&TFL_Part./&tflno..rtf"
style=t106326 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

title ;
footnote;
%let wd=0;
%let subpage=1;

%do j=1 %to &subpage;

%let maxpage=%eval(&page*&subpage);

%let npage=%eval(&subpage*&i+&j-&subpage);

```

```

data comp;
  set paging end=eof;
  where page=&i;

  /* Amend title as needed */
  _firtitl="Table 15.2.1.3 Summary of Protocol Deviations -
Safety Population";
  _upcas=(length(_firtitl)-
length(compress(_firtitl,'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
  len=&blankn.-length("(Page &NPAGE of &MAXpage)");
  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;
ods proclabel = ' ';
* 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 = '$' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
  column flag page roworder1 roworder2 /*rowtext1*/ sortord
('Significance/ Classification' param1)
  %IF &J=1 %THEN %DO; ("Sequence &linebot" ("THS 2.2 Menthol$-
mCC $(N=&trt1) &linebot" n1 p1) ("mCC -$THS 2.2 Menthol$(N=&trt2)
&linebot" n2 p2)
  ("THS 2.2 Menthol$- NRT gum $(N=&trt3) &linebot" n3 p3) /*
1) JR 11Aug2014 */
  ("NRT gum -$THS 2.2 Menthol$(N=&trt4) &linebot" n4 p4) ("
Exposed Not$Randomized$(N=&trt6) &linebot" N6 P6))

  ("Overall$Safety$(N=&trt99) &linebot" n99 p99)
%END; ;

  define flag          / order order = internal noprint;
  define page          / order order = internal noprint;
  define roworder1     / order order = internal noprint;
  define roworder2     / order order = internal noprint;
  define sortord       / order order = internal noprint;
  /*define rowtext      / group style={just=left
cellwidth=1.3cm}'Major/$minor'*/ /*'Major/$Minor '*'*/ /* 1) JR 11Aug2014
*/ /* 2) JR 23Sep2014 */

```

```

        define /*param*/PARAM1          / group style={just=left
cellwidth=3cm} style(header)={just=c} 'of Deviation'; /* 2) JR
23Sep2014 */
%if &j=1 %then %do;
        define n1                      / display style={just=d cellwidth=0.6cm}
style(header)={just=right};
        define p1                      / display style={just=left
cellwidth=1.3cm} style(header)={just=center};
        define n2                      / display style={just=d cellwidth=0.6cm}
style(header)={just=right};
        define p2                      / display style={just=left
cellwidth=1.3cm} style(header)={just=center};
        define n3                      / display style={just=d cellwidth=0.6cm}
style(header)={just=right};
        define p3                      / display style={just=left
cellwidth=1.3cm} style(header)={just=center};
        define n4                      / display style={just=d cellwidth=0.6cm}
style(header)={just=right};
        define p4                      / display style={just=left
cellwidth=1.3cm} style(header)={just=center};
        define n6                      / display style={just=d cellwidth=0.6cm}
style(header)={just=right};
        define p6                      / display style={just=left
cellwidth=1.3cm} style(header)={just=center};
        define n99                     / display style={just=d cellwidth=0.6cm}
style(header)={just=right};
        define p99                     / display style={just=left
cellwidth=1.3cm} style(header)={just=center};
        %END;

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

break after page / page;

compute after /*param*/ROWORDER1; /* 2) JR 23Sep2014 */
        line " ";
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 "&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: Exposed not randomized refers to all subjects
exposed to THS 2.2 Menthol or NRT gum 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 15.3.1.11';
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &npage of
&maxpage)" ;
        line "Program Run: &sysdate   &sysuserid   Program Status:
&status";
        endcomp;
run;
%end;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

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

ods listing;
proc printto print = "&table./t_15_02_01_03.lst" new;
run;

proc contents data = table.t_15_02_01_03 varnum;
run;
ods listing close;

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

```