

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106324;
%put NOTE: Client Protocol ID   : ZRHR-REXC-03-EU;
%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_jhardman;
%put NOTE: Creation Date        : 2014-07-28;
%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: 29Aug2014   JMH       1) Amended format as requested by the
client;
%put NOTE: 20Oct2014   KB        2) Removed zeroes if no data;
%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 adsl;
    set adam.adsl;
    where saffl = 'Y';
    attrib headtext1 length =$200.
                headorder1 length=8.;

    if missing(trt01a) or trt01a='Screen failure' then delete;
    if index(trt01a,'Enroll') then do;
        trt01an=98;
        trt01a='Exposed not randomized';
    end;
    headorder1=trt01an;
    headtext1=trt01a;
    output;
    trt01an=99;
    headorder1=99;
    trt01a='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';
    output;
    headorder1=2;
    headtext1='CC';
    output;
    headorder1=3;
    headtext1='SA';
    output;
    headorder1=98;
    headtext1='Exposed not randomized';
    output;
run;

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

data tot2;

```

```

merge tot(in=a) dumtrts(in=b);
by headorder1 headtext1;
if b and not a then count=0;
call symput('trt' || compress(put(headorder1,best.))),
compress(count));
run;

data addv;
set adam.addv;
where saffl = 'Y' and enrfl = 'Y';
if index(trta,'Enroll') then do;
    trtan=98;
    trta='Exposed not randomized';
end;
output;
trtan=99;
trta='Overall Safety';
output;
run;

data deviation02;
set addv;
attrib headtext1 length=$200.
        headorder1 length=8.;

headorder1=trtan;
headtext1=trta;
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=trtan;
    headtext1=trta;

    dvdecod=avalc;
    drop trtan trta;
run;

proc sql;
    create table results01 as
    select headorder1, headtext1, count(distinct usubjid) as treated
    from adsl
    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;

/* 1) START JMH 29Aug2014 */
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;
/* 1) END JMH 29Aug2014 */

data results03;
    merge results02_EXTRA(in=a) results01(keep=headorder1 headtext1
treated); /* 1) JMH 29Aug2014 */
    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(WHERE=(ROWORDER2 NE 0)));
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;
    set results06;
    length text text2 text3 $200. ;
    if (events = . and subjects = .) or missing(events) and
missing(subjects) then do;
        events    = .;
        subjects  = 0;
    end;
    percent = 100 * subjects / treated;

```

```

/*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) or events=0 then text2='';
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*/; /* 1) JMH 29Aug2014 */
rowtext='Major';
headorder1=1;
headtext1='THS 2.2';
output;
headorder1=2;
headtext1='CC';
output;
headorder1=3;
headtext1='SA';
output;
headorder1=98;
headtext1='Exposed not randomized';
output;
headorder1=99;
headtext1='Overall Safety';
output;
run;

data dumtrts_min; /*use this to output any columns for which n=0*/
attrib headtext1 length=$200.
rowtext length=$70.

```

```

                                headorder1 length=8.;
roworder1=/*2*/3; /* 2) JR 16Sep2014 */
roworder2=0/*1*/; /* 1) JMH 29Aug2014 */
rowtext='Minor';
headorder1=1;
headtext1='THS 2.2';
output;
headorder1=2;
headtext1='CC';
output;
headorder1=3;
headtext1='SA';
output;
headorder1=98;
headtext1='Exposed not randomized';
output;
headorder1=99;
headtext1='Overall Safety';
output;
run;

data dumtrts1;
    set dumtrts_maj dumtrts_min;
run;

proc sort data = dumtrts1;
    by headorder1 headtext1 roworder1 roworder2 rowtext;
run;

data results07a;
    merge results07(in=a) dumtrts1(in=b);
    by headorder1 headtext1 roworder1 roworder2 rowtext;
    if a or b;
    IF B AND NOT A THEN DO; /* 2) JR 16Sep2014 */
        TEXT='0';
        TEXT2='0';
    END;
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;

    /* 1) start JMH 29Aug2014 */
    /*    if missing(dvcat) then delete;    */

    /*    MISS=CMISS(N1, N2, N3, N98, N99);    */ /* 2) JR 16Sep2014 */
    /*    IF MISS=5 THEN DELETE;    */

    IF ROWORDER2=0 THEN PARAM=ROWTEXT;
    /* 1) end JMH 29Aug2014 */

    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 n1 label = " n"
           n2 label = " n"
           n3 label = " n"
           n98 label = " n"
           n99 label = " n"
           p1 label = '(%)'
           p2 label = '(%)'
           p3 label = '(%)'
           p98 label = '(%)'
           p99 label = '(%)'
           e1 label = "Events"
           e2 label = "Events"
           e3 label = "Events"
           e98 label = "Events"

```



```

        e99 label = "Events"
        rowtext label = "Deviation Class";

        attrib wrap length = $200
                PARAM1 LENGTH=$200.; /* 1) JMH 29Aug2014 */
        wrap = /*rowtext*/PARAM; /* 1) JMH 29Aug2014 */

/*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 0/*2*/ then do; /* 1) JMH
29Aug2014 */
    i=45; *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;
                    /*rowtext1*/PARAM1=substr(wrap,1,j-1) || "$n
$S={foreground=white} . $S={} " || substr(wrap,j+1); /* 1) JMH 29Aug2014
*/
                    fin=1;
                end;
                else j=j-1; *no space found - move back one character;
            end;
            nwraps=nwraps-1; *once this wrap is handled, move up a line
until all are handled (when nwraps = 0);
        end;
        /*rowtext1*/PARAM1='$S={foreground=white} . $S={} ' ||
left(trim(PARAM1/*rowtext1*/)); /* 1) JMH 29Aug2014 */
    end;
    else do;
        /*rowtext1*/PARAM1='$S={foreground=white} . $S={} ' ||
left(trim(PARAM/*rowtext*/)); /* 1) JMH 29Aug2014 */
    end;

end;
else do; /*rowtext1=rowtext*/PARAM1=PARAM; end; /* 1) JMH 29Aug2014 */

array a [4] n1 n2 n3 n98;
do i=1 to 4;
    if missing(a[i]) then a[i] ='0';
end;

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

        flag = 1;
run;

```

```

/* 2) START KB 20Oct2014 */
PROC SORT DATA=LABELS (WHERE=(ROWORDER2=0)) OUT=BLANKS;
    BY ROWORDER1 ROWORDER2 ROWTEXT DVCAT PARAM;
RUN;

DATA BLANKS1;
    SET BLANKS;
        IF (N1='0' OR MISSING(N1)) THEN DEL1=1;
        IF (N2='0' OR MISSING(N2)) THEN DEL2=1;
        IF (N3='0' OR MISSING(N3)) THEN DEL3=1;
        IF (N98='0' OR MISSING(N98)) THEN DEL98=1;
    KEEP ROWORDER1 ROWTEXT DEL;
RUN;

PROC SORT DATA=LABELS;
    BY ROWORDER1 ROWTEXT;
RUN;

DATA BLANKS2;
    MERGE LABELS BLANKS1;
    BY ROWORDER1 ROWTEXT ;
RUN;

DATA BLANKS3;
    SET BLANKS2;

    ARRAY DEL [4] DEL1 DEL2 DEL3 DEL98;
    ARRAY NUM [4] N1 N2 N3 N98;

    DO I=1 TO 4;
        IF DEL[I]=1 AND PARAM NOT IN ('Major' 'Minor') THEN NUM[I]='';
    END;
RUN;
/* 2) END KB 20Oct2014 */

proc sql noprint;

create table table.t_15_02_01_03 as
select /*rowtext, rowtext1,*/PARAM, PARAM1, dvcat, n1, n2, n3, n98, n99,
p1, p2, p3, p98, p99 /* 1) JMH 29Aug2014 */
from /*labels*/BLANKS3 /* 2) KB 20Oct2014 */
order by roworder1, roworder2, sortord;

quit;

data paging;
    set /*labels*/BLANKS3; /* 2) KB 20Oct2014 */
    by roworder1 roworder2 sortord dvcat param;
        if first.roworder1 AND LN GT 5 then ln=1; /*Amend to look
presentable, and avoid page overflows*/ /* 1) JMH 29Aug2014 */
    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.tl06324 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106324/TFL/&TFL_Part./&tflno..rtf"
style=tl06324 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

title ;
footnote;
%let wd=0;

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 &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;
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 /*('Major/' rowtext1)*/
sortord ('Significance/ Classification' PARAM1/*param*/) /* 1) JMH
29Aug2014 */
    ("THS 2.2 $(N=&trt1) &linebot" n1 p1) ("CC$(N=&trt2) &linebot" n2
p2)
    ("SA $(N=&trt3) &linebot" n3 p3) ("Exposed
Not$Randomized$(N=&trt98) &linebot" n98 p98)
    ("Overall$Safety$(N=&trt99) &linebot" n99 p99); ;

    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 rowtext1     / group style={just=left cellwidth=1.3cm}
style(header)={just=c} 'minor'; */ /* 1) JMH 29Aug2014 */
    define /*param*/PARAM1 / group style={just=left
cellwidth=3cm} style(header)={just=c} 'of Deviation'; /* 1) JMH
29Aug2014 */
    define n1            / display style={just=d cellwidth=0.3cm}
style(header)={just=r} ;
    define p1            / display style={just=c cellwidth=1.3cm}
style(header)={just=c} ;
    define n2            / display style={just=d cellwidth=0.3cm}
style(header)={just=r} ;
    define p2            / display style={just=c cellwidth=1.3cm}
style(header)={just=c} ;
    define n3            / display style={just=d cellwidth=0.3cm}
style(header)={just=r} ;
    define p3            / display style={just=c cellwidth=1.3cm}
style(header)={just=c} ;
    define n98           / display style={just=d cellwidth=0.3cm}
style(header)={just=r} ;
    define p98           / display style={just=c cellwidth=1cm}
style(header)={just=c} ;
    define n99           / display style={just=d cellwidth=0.5cm}
style(header)={just=r} ;
    define p99           / display style={just=c cellwidth=1.3cm}
style(header)={just=c} ;

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

    break after page / page;

    compute after /*param*/ROWORDER1; /* 1) JMH 29Aug2014 */
        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: CC = Conventional cigarettes; SA = Smoking
abstinence; THS = Tobacco Heating System';
  line 'Note: Exposed Not Randomized refers to all subjects
exposed to THS 2.2 but not randomized. Overall Safety refers to all
subjects exposed to THS 2.2.';
  line 'Note: Percentages are based on the number of subjects
indicated in the column header (N).';
  line ' ';
  LINE 'Appendix 15.3.1.10';
  line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of &page)"
;
  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);

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 ;
*=====;

```