

```

%_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_adv4.sas;
%put NOTE: Purpose              : table of adverse events by preferred
term;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADAE;
%put NOTE: Output               : t_15_2_6_2_3(ae);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jhardman;
%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:
=====;
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_06_02_03(ae);

/* 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';
    if missing(trtseqa) then delete;
    if index(trtseqa,'Enroll') then do;
        trtseqa=6;
        trtseqa='Exposed not randomized';
    end;
    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=6;
    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;

/*Now bring in AE data*/
data ae;
    attrib aebodsys length=$200.;
    set adam.adae;
    where saffl='Y' and anyae1='Y' and anl01fl='Y';
    if missing(trtseqan) then delete;
    if missing(aebodsys) then aebodsys='Missing';
    if index(trtsega,'Enroll') then do;
        trtseqan=6;
        trtsega='Exposed not randomized';
    end;
    headorder1=trtseqan;
    headtext1=trtsega;
    output;
    headorder1=99;
    headtext1='Overall Safety';
    output;
run;

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

data adverse02;
    set ae;
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 adverse03;
    set adverse02;
    output;
    ae1decod='Any Adverse events';
    output;
run;

/** Number of Ae's overall **/
proc freq data=adverse03 noprint;
    tables headorder1*headtext1*ae1decod / out=ovall(rename=(count=tot)
drop=percent);
run;

/** getting number of subjects studied ***/
proc sort data=adverse03 out=nae3 nodupkey;
    by headorder1 headtext1 ae1decod subjidn;
run;

```

```

proc freq data=nae3 noprint;
  tables headorder1*headtext1*aedecod/ out=novall(rename=(count=ntot)
drop=percent);
run;

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

/** number of subjects and aes in overall **/
data overall;
  set otot;
  if aedecod='Any Adverse events' then sort2=1;
  else sort2=2;
  sort3=0;
run;

data body2;
  set overall;
  if aedecod='Any Adverse events' then do; sort2=1; sort3=0;
end;
  else do; sort2=2; sort3=0; end;
run;

data all;
  set body2;
run;

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

data format;
  merge all2(in=a) dumtrts tot;
  by headorder1 headtext1;
  if not a then do;
    sort2=1;
    sort3=0;
    aedecod='Any Adverse events';
    ntot=0;
  end;
run;

data format2;
  set format;
  attrib text text2 text3 format=$200.;
  /* 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='';
        else text=put(ntot,3.);

        /*% value*/
        if missing(percent) or percent=0 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(tot) or tot=0 then text2='';
        else text2=compress(put(tot,3.));

        drop percent;

run;

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

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

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

/*Transpose n values*/
proc transpose data=format3 out=nformat prefix=n;
  by sort2 aedecod sort3;
  var text;
  id headorder1;
  idlabel headtext1;
run;

/*Transpose % values*/
proc transpose data=format3 out=performat prefix=p;
  by sort2 aedecod sort3;
  var text3;
  id headorder1;
  idlabel headtext1;
run;

/*Transpose events values*/
proc transpose data=format3 out=eformat prefix=e;
  by sort2 aedecod sort3 ;
  var text2;
  id headorder1;
  idlabel headtext1;
run;

        data tformat;
        merge nformat eformat performat;
        by sort2 aedecod sort3;

```

```

        /*Create numeric variables for sorting*/
        n99n=input(n99,8.);
        e99n=input(e99,8.);
run;

/*Sort preferred terms in order of most frequent*/
proc sort data=tformat out=tsort2;
    by sort2 sort3 descending n99n descending e99n;
run;

data sorting;
    set tsort2;
    by sort2 sort3 descending n99n descending e99n aedecod;
    if first.aedecod then odd+1; /*Odd now sorts preferred terms in the
required order*/
run;

data final2;
    set sorting;
    attrib column format=$70.;

    if sort2=2 then column=trim(aedecod);
    else if sort2=1 then do; column='Any adverse events'; end;
run;

proc sort data=final2;
    by sort2 odd;
run;

data labels;
    set final2;
    attrib n1 label = " n"
           n2 label = " n"
           n3 label = " n"
           n4 label = " n"
           n6 label = " n"
           n99 label = " n"
           p1 label = '(%)'
           p2 label = '(%)'
           p3 label = '(%)'
           p4 label = '(%)'
           p6 label = '(%)'
           p99 label = '(%)'
           e1 label = "Events"
           e2 label = "Events"
           e3 label = "Events"
           e4 label = "Events"
           e6 label = "Events"
           e99 label = "Events";
run;

data final4;
    set labels;

```

```

        array a [4] n1 n2 n4 n99;
        array c [4] e1 e2 e4 e99;
        do i=1 to 4;
            if missing(a[i]) then a[i] ='0';
            if length(left(strip(c[i])))=1 then c[i]=
'|S={foreground=white}.|S={}' || left(strip(c[i]));
        end;

            flag=1;

        run;

proc sql noprint;
    create table table.T_15_02_06_02_03 as
    select column, aedecod, n1, p1, e1, n2, p2, e2, n3, p3, e3, n4, p4,
e4, n6, p6, e6, n99, p99, e99
    from final4
    order by sort2, odd;
quit;

data paging;
    set final4;
    by sort2 odd ;
    if ln gt 11 then ln=1;
    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=70, halfblnk=N);

%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/* contents*/
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;
ods proclabel = ' ';

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

    /* Amend title as needed */
    _firtitl="Table 15.2.6.2.3 Summary of Adverse Events by
Preferred Term - 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 = '$' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
column flag page sort2 odd column ("Sequence &linebot" ("THS 2.2
Menthol$- mCC $(N=&trt1) &linebot" n1 p1 e1)

("mCC -$THS 2.2 Menthol$(N=&trt2) &linebot" n2 p2 e2)

("THS 2.2 Menthol$- NRT gum $(N=&trt3) &linebot" n3 p3 e3)

("NRT gum -$THS 2.2 Menthol$(N=&trt4) &linebot" n4 p4 e4)

("Exposed Not$Randomized$(N=&trt6) &linebot" n6 p6 e6))

("Overall$Safety$(N=&trt99) &linebot" n99 p99 e99); ;
    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 column      / group style={just=left cellwidth=3.2cm}
style(header)={just=center}"Preferred Term";
        define n1      / display style={just=d cellwidth=0.3cm}
style(header)={just=right};
        define p1      / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
        define n2      / display style={just=d cellwidth=0.3cm}
style(header)={just=right};
        define p2      / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
        define n3      / display style={just=d cellwidth=0.3cm}
style(header)={just=right};
        define p3      / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
        define n4      / display style={just=d cellwidth=0.3cm}
style(header)={just=right};
        define p4      / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
        define n6      / display style={just=d cellwidth=0.2cm}
style(header)={just=right};
        define p6      / display style={just=d cellwidth=1cm}
style(header)={just=center};
        define n99      / display style={just=d cellwidth=0.3cm}
style(header)={just=right};
        define p99      / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
        define e1      / display
style={cellwidth=1.2cm} style(header)={just=center};
        define e2      / display
style={cellwidth=1.2cm} style(header)={just=center};
        define e3      / display
style={cellwidth=1.2cm} style(header)={just=center};
        define e4      / display
style={cellwidth=1.2cm} style(header)={just=center};
        define e6      / display
style={cellwidth=1.2cm} style(header)={just=center};
        define e99      / display
style={cellwidth=1.2cm} style(header)={just=center};

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

compute after sort2;
        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 15.3.6.1.1';
        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_06_02_03.lst" new;
run;

proc contents data = table.T_15_02_06_02_03 varnum;
run;
ods listing close;

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

```