

```

%_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_conatc.sas;
%put NOTE: Purpose              : table of concomitant medications by
ATC;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADCM ADAM.ADSL;
%put NOTE: Output               : t_15_2_6_9_1(conatc);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_kbooth;
%put NOTE: Creation Date        : 2014-04-16;
%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: 06Aug2014   JMH       1) Applied formatting updates;
%put NOTE: 11Aug2014   JMH       2) Amended table and footnote;
%put NOTE: ;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

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

%let tflno=T_15_02_06_09_01(conatc);

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

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(trtsega) then delete;
  if index(trtsega,'Enroll') then do;
    trtsega=6;
    trtsega='Exposed not randomized';
  end;
  headorder1=trtsega;
  headtext1=trtsega;
  output;
  trtsega=99;
  headorder1=99;
  trtsega='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 sort data=adsl1 out=adsl; by headorder1 headtext1; 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 a or b;
  if b and not a then count=0;

```

```

        call symput('trt' || compress(put(headorder1,best.)),
compress(count));
run;

%macro test;

%if %sysfunc(exist(adam.adcm))=0 %then %do;
data paging;
    page=1; flag=1; ln=1; sort2=.; odd=.; sort3=.; odd2=.; column='';
n1=''; p1=''; e1=''; n2=''; p2=''; e2='';
    n3=''; p3=''; e3=''; n4=''; p4=''; e4=''; n6=''; p6=''; e6=''; n99='';
p99=''; e99='';
    output;
    call symput("page",'1');

        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;
%end;

%else %if %sysfunc(exist(adam.adcm)) %then %do;
%put "USER WARN" "ING: ADCM exists, update code.";

data cml;
    set adam.adcm;
    where anycmfl='Y' and cmfl='Y' and saffl='Y';
    if missing(trtseqan) then delete;
    if index(trtseqa,'Enroll') then delete;
run;

data cm;
    set cml;
    headorder1=trtseqan;
    headtext1=trtseqa;
    output;
    headorder1=99;
    headtext1='Overall Safety';

```

```

        output;
run;

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

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

data cm02;
    merge cm(in=a) adsl(in=b); /*Only keep subjects with Conmeds*/
    by headorder1 headtext1 usubjid;
    if a and b;
run;

proc sort data=cm02; by headorder1 headtext1; run;

data cm03;
    set cm02;
    output;
    cmatc1='Any Medication';
    output;
run;

/** Number of CMs overall **/
proc freq data=cm03 noprint;
tables headorder1*headtext1*cmatc1 / out=ovall(rename=(count=tot)
drop=percent);
run;

/** getting number of subjects studied **/
proc sort data=cm03 out=ncm3 nodupkey;
    by headorder1 headtext1 cmatc1 subjidn;
run;

proc freq data=ncm3 noprint;
    tables headorder1*headtext1*cmatc1 / out=novall(rename=(count=ntot)
drop=percent);
run;

data otot;
    merge ovall novall;
    by headorder1 headtext1 cmatc1 ;
run;

/** number of subjects and CMs in overall **/
data overall;
set otot;
    /*Sort2 will sort CLASS 1, as we want Any Medication to be the
first row in the table,*/
    /*we set sort2 to=1 for Any Medication. For all other class 1s, it
=2*/
    if cmatc1='Any Medication' then sort2=1;
    else sort2=2;

```

```

run;

/* Ordering the body systems with most frequent first*/

    /*Body systems ordered by highest number of events*/
proc freq data=cm03 noprint;
    tables headorder1*headtext1*cmatc1 / out=tcmbod(rename=(count=tot)
drop=percent);
run;

    /* BODY SYSTEMS ORDERED BY MOST SUBJECTS */
proc sort data=cm03 out=ncm2 nodupkey;
    by headorder1 headtext1 cmatc1 subjdn;
run;

proc freq data=ncm2 noprint;
    tables headorder1*headtext1*cmatc1 / out=cmbod(rename=(count=ntot)
drop=percent);
run;

    /*Combine number of events and subjects for each body system*/
data body;
    merge tcmbod cmbod;
    by headorder1 headtext1 cmatc1;
run;

data body2;
    set body;
    /*Sort 2 sorts class 1, sort3 will sort pclass 2, so here
sort3 can=0 as class 2 isnt included yet*/
    if cmatc1='Any Medication' then do; sort2=1; sort3=0; end;
    else do; sort2=2; sort3=0; end;
run;

/*Sorting each class 1 by class 2*/

proc freq data=cm03 noprint;
tables headorder1*headtext1*cmatc1*cmatc2 / out=preft(rename=(count=tot)
drop=percent);
run;

/** getting number of subjects studied **/
proc sort data=cm03 out=npcm2 nodupkey;
    by headorder1 headtext1 cmatc1 cmatc2 subjdn;
run;

proc freq data=npcm2 noprint;
tables headorder1*headtext1*cmatc1*cmatc2 /
out=npreftr(rename=(count=ntot) drop=percent);
run;

data prefterm;
merge preft npreftr;

```

```

by headorder1 headtext1 cmatc1 cmatc2;
run;

data cmdecod;
set prefterm;

    /*Sort 2 sorts bclass 1, sort3 will sort class 2.*/
    /*Here sort3 orders class2 by most events then most subjects,
within each bclass 1*/
    if cmatc1='Any Medication' then do; sort2=1; sort3=0; end;
    else do; sort2=2; sort3=1; end;
run;

data all;
    set overall body2 cmdecod;
run;

proc sort data=all out=all2;
    by headorder1 headtext1 sort2 cmatc1 cmatc2;
run;

data format;
merge all2(in=a) dumtrts tot;
by headorder1 headtext1;
    if not a then do;
        sort2=1;
        sort3=0;
        cmatc1='Any Medication';
        dumflag=1;
    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 percent=100 then text3='(100 %)';
    else if percent=0 or missing(percent) then text3='';
    else if percent ge 10 then text3='( ' ||
left(compress(put(percent,8.1))) || '%' )';
    else if percent lt 10 then text3='( ' ||
left(compress(put(percent,8.1))) || '%' )';

    /*events value*/
    if missing(tot) then text2='0';

```

```

else text2=compress(put(tot,3.));

/*Any Medication isn't broken down into classes so some rows
can be deleted*/
if dumflag ne 1 and cmatc1='Any Medication' and (not
missing(cmatc2) or missing(sort3)) then delete;
drop percent dumflag;
run;

proc sort data=format2; by headorder1 headtext1 sort2 cmatc1 sort3
cmatc2; run;

proc sort data=format2 out=format3; by sort2 cmatc1 sort3 cmatc2; run;

proc transpose data=format3 out=nformat prefix=n;
by sort2 cmatc1 sort3 cmatc2;
var text;
id headorder1;
idlabel headtext1;
run;

proc transpose data=format3 out=eformat prefix=e;
by sort2 cmatc1 sort3 cmatc2;
var text2;
id headorder1;
idlabel headtext1;
run;

proc transpose data=format3 out=pformat prefix=p;
by sort2 cmatc1 sort3 cmatc2;
var text3;
id headorder1;
idlabel headtext1;
run;

data tformat;
merge nformat eformat pformat;
by sort2 cmatc1 sort3 cmatc2;

if missing(sort3) then delete;
run;

/*Order bodysystems by number of events and subjects*/
data bodsysort;
set tformat;
where sort3=0;
/*create numeric variables for sorting*/
e99_n=input(e99,8.);
n99_n=input(n99,8.);
run;

```

```

proc sort data=bodsyssort;
    by sort2 sort3 descending e99_n descending n99_n;
run;

data bodsyssort2;
    set bodsyssort;
    odd+1;
    keep sort2 cmatc1 odd;
run;

/*Merge this back onto tformat to include the variable odd which
orders the CLASS 1S*/
proc sort data=tformat; by sort2 cmatc1 sort3 cmatc2; run;
proc sort data=bodsyssort2; by sort2 cmatc1; run;

data systems;
    merge tformat bodsyssort2;
    by sort2 cmatc1 ;
run;

/*Order class 2S within each bclass 1 by number of events and subjects*/
data prefsort;
    set systems;
    where sort3=1;
    /*create numeric variables for sorting*/
    e99_n=input(e99,8.);
    n99_n=input(n99,8.);
run;

proc sort data=prefsort;
    by odd descending e99_n descending n99_n;
run;

data prefsort2;
    set prefsort;
    odd2+1;
    keep sort2 cmatc1 sort3 cmatc2 odd odd2;
run;

/*Merge this back onto systems to include the variable odd2 which
orders the CLASS 2S*/
proc sort data=systems; by sort2 cmatc1 sort3 cmatc2; run;
proc sort data=prefsort2; by sort2 cmatc1 sort3 cmatc2; run;

data final;
    merge systems prefsort2;
    by sort2 cmatc1 sort3 cmatc2;
run;

/*now the data can be sorted using odd and odd2*/
proc sort data=final;
    by sort2 odd odd2;
run;

```



```

data final3;
    set final;
    attrib column column1 wrap format=$400.;
    if sort2=1 then do; column1='Any Medication'; odd2=0; end;
    else if sort2=2 then do;
        if sort3=0 then column1=left(trim(propcase(cmatc1)));
        else if sort3=1 then column1=left(trim(propcase(cmatc2)));
    end;

wrap = column1;

if sort2=2 and sort3=1 then do;
i=14; *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;
        test=j;
        do while(fin=0 AND J GT 1);
            if substr(wrap,j,1)=' ' then do;
                wrap=substr(wrap,1,j-1) || " |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;

else column=column1;

run;

proc sort data=final3;
    by sort2 odd odd2;
run;

data labels;
    set final3;
    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;

        if missing(n1) then n1='0';
        if missing(e1) then e1='0';
        if missing(n2) then n2='0';
        if missing(e2) then e2='0';
        if missing(n3) then n3='0';
        if missing(e3) then e3='0';
        if missing(n4) then n4='0';
        if missing(e4) then e4='0';
        if missing(n99) then n99='0';
        if missing(e99) then e99='0';

run;

proc sql noprint;

create table table.t_15_02_06_09_01 as
select cmatc1, cmatc2, column, column1, n1, n2, n3, n4, n6, n99, e1, e2,
e3, e4, e6, e99, p1, p2, p3, p4, p6, p99
from final4
order by sort2, ODD, odd2;

quit;

data paging;
set final4;
by sort2 odd odd2;

flag=1;

    if (ln gt 2 and first.odd) or ln gt 5 then ln=1;
else ln+1;

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

%end;

```

```

%mend test;
%test;

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;

%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;

data comp;
    set paging end=eof;
    where page=&i;
    if missing(column) then call symput('NOOBS',1);

    /* Amend title as needed */
    _firtitl="Table 15.2.6.9.1 Summary of Concomitant Medication
by Anatomical Therapeutic Classes (ATC) 1 and 2 - 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;
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 headsip missing nowd split = '$' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;;
    column flag page sort2 odd odd2 ("ATC1" 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)*/

("Exposed
Not$Randomized$(N=&trt6) &linebot" n6 p6 e6)) /* 1) JMH 06Aug2014 */

("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 odd2          / order order=internal noprint;

    define column        / group style={just=left cellwidth=3.5cm}
"|~ ATC2" style(header)={just=center};
    define n1            / display style={just=d cellwidth=0.3cm}
style(header)={just=center};
    define n2            / display style={just=d cellwidth=0.3cm}
style(header)={just=center};
    define n3            / display style={just=d cellwidth=0.3cm}
style(header)={just=center};
    define n4            / display style={just=d cellwidth=0.3cm}
style(header)={just=center};
    define n6            / display style={just=d cellwidth=0.3cm}
style(header)={just=center};
    define n99           / display style={just=d cellwidth=0.4cm}
style(header)={just=center};
    define p1            / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
    define p2            / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
    define p3            / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
    define p4            / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
    define p6            / display style={just=d cellwidth=0.6cm}
style(header)={just=center};

```

```

        define p99                / display style={just=d cellwidth=1.2cm}
style(header)={just=center};
        define e1                / display style={cellwidth=1.15cm
pretext="\tqdec\tx500 "} style(header)={just=center};
        define e2                / display style={cellwidth=1.15cm
pretext="\tqdec\tx500 "} style(header)={just=center};
        define e3                / display style={cellwidth=1.15cm
pretext="\tqdec\tx500 "} style(header)={just=center};
        define e4                / display style={cellwidth=1.15cm
pretext="\tqdec\tx500 "} style(header)={just=center};
        define e6                / display style={cellwidth=1.15cm
pretext="\tqdec\tx500 "} style(header)={just=center};
        define e99               / display style={cellwidth=1.15cm
pretext="\tqdec\tx500 "} style(header)={just=center};

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

        break after page / page;

%IF &NOOBS. NE 1 %THEN %DO; /* 2) JMH 11Aug2014 */
        compute after sort2;
                line " ";
        endcomp;
%END; /* 2) JMH 11Aug2014 */

        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 " ";*/ /* 2) JMH 11Aug2014 */
/*                line "No concomitant medication data";*/
                line "No concomitant medication reported"; /* 1) JMH 06Aug2014 */
                line " ";
        %end;
/*                line "&linebot" ;*/ /* 1) JMH 06Aug2014 */
        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."}; /* 1) JMH 06Aug2014 */
/*                line 'Note: "Exposed not randomized" refers to all subjects
exposed to THS 2.2 Menthol or NRT gum but not randomized. The Overall
Safety refers to all subjects exposed to THS 2.2 Menthol or NRT gum.'; */

```

```

/*          line 'Note: mCC = Menthol conventional cigarettes; NRT gum =
Nicotine replacement therapy gum; THS = Tobacco Heating System.';*/
/*          line 'Note: Percentages are based on the number of subjects
in the column header (N).';*/
/*line 'Note: mCC = menthol conventional cigarettes; NRT gum
= Nicotine Replacement Therapy gum; THS = Tobacco Heating System.';*/ /*
1) JMH 06Aug2014 */
    line 'Note: mCC = menthol conventional cigarettes; NRT gum =
Nicotine Replacement Therapy gum; THS = Tobacco Heating System.'; /* 2)
JMH 11Aug2014 */
    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.'; /* 1)
JMH 06Aug2014 */
    line 'Note: Percentages are based on the number of subjects
indicated in the column header (N).'; /* 1) JMH 06Aug2014 */
    line ' ';
    line 'Appendix 15.3.6.3.2';
    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=);
%macro test2;
%if %sysfunc(exist(adam.adcm)) %then %do;

ods listing;
proc printto print = "&table./t_15_02_06_09_01.lst" new;
run;

proc contents data = table.t_15_02_06_09_01 varnum;
run;
ods listing close;
%end;
%mend test2;
%test2;

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

```

