

```

%_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-02-JP;*/ ZRHM-PK-05-JP; /* 4)
AOB 12Aug2014 */
%put NOTE: Program Name        : t_adv1.sas;
%put NOTE: Purpose             : table of adverse events;
%put NOTE: ;
%put NOTE: Input Data          : ADAM.ADAE;
%put NOTE: Output              : t_15_2_6_1(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: 12Aug2014  AOB        1) Column headers and widths amended;
%put NOTE: 12Aug2014  AOB        2) Paging amended;
%put NOTE: 12Aug2014  AOB        3) Wrapping of ROWTEXT amended;
%put NOTE: 12Aug2014  AOB        4) Program header amended;
%put NOTE: 12Aug2014  AOB        5) Amendments 1) to 3) reversed;
%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_01(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(trtseqan) then delete;
    if index(trtseqa,'Enroll') then do;
        trtseqan=6;
        trtseqa='Exposed not randomized';
    end;
    headorder1=trtseqan;
    headtext1=trtseqa;
    output;
    trtseqan=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;
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;
    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(trtseqa,'Enroll') then do;
        trtseqan=6;
        trtseqa='Exposed not randomized';
    end;
    headorder1=trtseqan;
    headtext1=trtseqa;
    output;
    headorder1=99;
    headtext1='Overall Safety';
    output;
run;

```

```

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

```

```

data adverse02;
    set ae;
run;

```

```

proc sort data=adverse02; 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 adverse03;
    set adverse02;
    by headorder1 headtext1;
    output;

```

```

    if first.headtext1 then do;
        subjid = .;
        aeterm='';
        output;
    end;

```

```

run;

* Create values for table rows;
data adverse04;
  set adverse03;
  length rowtext $200;
  * All adverse events;
  roworder1 = 2;
  roworder2 = 1;
  rowtext = 'Adverse events (AE)';
  output;
  * Serious adverse events - header;
  roworder1 = 3;
  roworder2 = 1;
  rowtext = 'AE serious';
  output;
  * Serious adverse events - missing;
  roworder1 = 3;
  roworder2 = 2;
  rowtext = '$S={foreground=white} . $S={} Missing';
  if subjid = . or aeser = '' then output;
  * Serious adverse events - yes;
  roworder1 = 3;
  roworder2 = 3;
  rowtext = '$S={foreground=white} . $S={} Yes';
  if subjid = . or aeser in('Y' 'YES' 'Yes') then output;
  * Serious adverse events - no;
  roworder1 = 3;
  roworder2 = 4;
  rowtext = '$S={foreground=white} . $S={} No';
  if subjid = . or aeser in('N' 'NO' 'No') then output;
  *Related to IP - header;
  roworder1 = 4;
  roworder2 = 1;
  rowtext = 'AE related to IP';
  output;
  *Related to IP - missing;
  roworder1 = 4;
  roworder2 = 2;
  rowtext = '$S={foreground=white} . $S={} Missing';
  if subjid = . or aerell = '' then output;
  *Related to IP - not related;
  roworder1 = 4;
  roworder2 = 3;
  rowtext = '$S={foreground=white} . $S={} Not related';
  if subjid = . or aerell in('N' 'NO' 'NOT RELATED' 'No') then output;
  *Related to IP - related;
  roworder1 = 4;
  roworder2 = 4;
  rowtext = '$S={foreground=white} . $S={} Related';
  if subjid = . or aerell in('Y' 'YES' 'RELATED' 'Yes') then output;
  *Related to IP - related - missing;
  roworder1 = 4;
  roworder2 = 5;

```

```

        rowtext = '$S={foreground=white} . $S={} Missing';
    if subjid = . or (aerel1 in('Y' 'YES' 'RELATED' 'Yes') and aeexpec =
'') then output;
        *Related to IP - related - expected;
    roworder1 = 4;
    roworder2 = 6;
    rowtext = '$S={foreground=white} . $S={} Expected';
    if subjid = . or (aerel1 in('Y' 'YES' 'RELATED' 'Yes') and aeexpec
in('Y' 'YES' 'Yes')) then output;
        *Related to IP - related - not expected;
    roworder1 = 4;
    roworder2 = 7;
    rowtext = '$S={foreground=white} . $S={} Not expected';
    if subjid = . or (aerel1 in('Y' 'YES' 'RELATED' 'Yes') and aeexpec
in('N' 'NO' 'No')) then output;
        *Related to NRT gum;
    roworder1 = 5;
    roworder2 = 1;
    rowtext = 'AE related to NRT gum';
    output;
        *Related to NRT gum - missing;
    roworder1 = 5;
    roworder2 = 2;
    rowtext = '$S={foreground=white} . $S={} Missing';
    if subjid = . or aerel2 = '' then output;
        *Related to NRT gum - not related;
    roworder1 = 5;
    roworder2 = 3;
    rowtext = '$S={foreground=white} . $S={} Not related';
    if subjid = . or aerel2 in('N' 'NO' 'NOT RELATED' 'No') then output;
        *Related to NRT gum - related;
    roworder1 = 5;
    roworder2 = 4;
    rowtext = '$S={foreground=white} . $S={} Related';
    if subjid = . or aerel2 in('Y' 'YES' 'RELATED' 'Yes') then output;
        *Related to NRT gum - related - missing;
    roworder1 = 5;
    roworder2 = 5;
    rowtext = '$S={foreground=white} . $S={} Missing';
    if subjid = . or (aerel2 in('Y' 'YES' 'RELATED' 'Yes') and aeexpec1 =
'') then output;
        *Related to NRT gum - related - expected;
    roworder1 = 5;
    roworder2 = 6;
    rowtext = '$S={foreground=white} . $S={} Expected';
    if subjid = . or (aerel2 in('Y' 'YES' 'RELATED' 'Yes') and aeexpec1
in('Y' 'YES' 'Yes')) then output;
        *Related to NRT gum - related - not expected;
    roworder1 = 5;
    roworder2 = 7;
    rowtext = '$S={foreground=white} . $S={} Not expected';
    if subjid = . or (aerel2 in('Y' 'YES' 'RELATED' 'Yes') and aeexpec1
in('N' 'NO' 'No')) then output;
        *Related to study procedure;

```

```

roworder1 = 6;
roworder2 = 1;
rowtext = 'AE related to study procedure';
output;
*Related to study procedure - missing;
roworder1 = 6;
roworder2 = 2;
rowtext = '$S={foreground=white} . $S={} Missing';
if subjid = . or aerelsp = '' then output;
*Related to study procedure - not related;
roworder1 = 6;
roworder2 = 3;
rowtext = '$S={foreground=white} . $S={} Not related';
if subjid = . or aerelsp in('N' 'NO' 'NOT RELATED' 'No') then output;
*Related to study procedure - related;
roworder1 = 6;
roworder2 = 4;
rowtext = '$S={foreground=white} . $S={} Related';
if subjid = . or aerelsp in('Y' 'YES' 'RELATED' 'Yes') then output;
* AE severity;
roworder1 = 7;
roworder2 = 1;
rowtext = 'AE severity';
output;
* Severity for all adverse events - mild;
roworder1 = 7;
roworder2 = 3;
rowtext = '$S={foreground=white} . $S={} Mild';
if subjid = . or aesev = 'MILD' then output;
* Severity for all adverse events - moderate;
roworder1 = 7;
roworder2 = 4;
rowtext = '$S={foreground=white} . $S={} Moderate';
if subjid = . or aesev = 'MODERATE' then output;
* Severity for all adverse events - severe;
roworder1 = 7;
roworder2 = 5;
rowtext = '$S={foreground=white} . $S={} Severe';
if subjid = . or aesev = 'SEVERE' or missing(aesev) then output;
*Action taken due to AE;
roworder1= 8;
roworder2=1;
rowtext='Action taken due to AE';
output;
*Action taken - discontinuation from study;
roworder1 = 8;
roworder2 = 2;
rowtext = '$S={foreground=white} . $S={} Discontinuation $n
$S={foreground=white} . $S={} from study';
if subjid = . or anl02fl = 'Y' then output;
*Action taken - related to product use;
roworder1 = 8;
roworder2 = 3;

```

```

rowtext = /*'$S={foreground=white} . $S={} Related to product
use'*/'$S={foreground=white} . $S={} Related to product $n
$S={foreground=white} . $S={} use'; /* 3) AOB 12Aug2014 */ /* 5) AOB
12Aug2014 */
if subjid = . or anl03fl = 'Y' then output;
    *Action taken -treatment given;
roworder1 = 8;
roworder2 = 4;
rowtext = '$S={foreground=white} . $S={} Treatment given';
if subjid = . then output;
    *Action taken -treatment given - missing;
roworder1 = 8;
roworder2 = 5;
rowtext = '$S={foreground=white} . $S={} Missing';
if subjid = . or aecontrt = '' then output;
    *Action taken -treatment given - yes;
roworder1 = 8;
roworder2 = 6;
rowtext = '$S={foreground=white} . $S={} Yes';
if subjid = . or aecontrt = 'Y' then output;
    *Action taken -treatment given - no;
roworder1 = 8;
roworder2 = 7;
rowtext = '$S={foreground=white} . $S={} No';
if subjid = . or aecontrt = 'N' then output;
    *Action taken - Other action taken;
roworder1 = 8;
roworder2 = 8;
rowtext = '$S={foreground=white} . $S={} Other action taken';
if subjid = . or anl05fl = 'Y' then output;
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=adverse04 out=adverse04_a ; by headorder1 headtext1
roworder1 roworder2 rowtext usubjid aeterm; run;

```

```

proc sql;
    create table results02 as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(aeterm) as events
    from adverse04_a
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;

```

```

proc sql;
    create table results02_allaes as

```

```

    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(aeterm) as events,
        count(distinct subjid) as subjects
    from adverse04_a(where=(roworder1 in(2 8)))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;

/*SERIOUS*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=3 and
roworder2 not in(1)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aesern aeser aseq) out=ser;
    by headorder1 headtext1 usubjid roworder2;
run;

proc sort data=ser(keep=headorder1 headtext1 roworder1 roworder2 rowtext
usubjid subjid aebodsys aeterm aesern aeser aseq) nodupkey out=ser1;
    by headorder1 headtext1 usubjid roworder2;
run;

data ser2;
    set ser1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    serflag=1;
run;

/*RELATED TO IP*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=4 and
roworder2 not in(1 5 6 7)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aerelln aerell aseq) out=relip;
    by headorder1 headtext1 usubjid descending roworder2;
run;

proc sort data=relip(keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aerelln aerell aseq) nodupkey
out=relip1;
    by headorder1 headtext1 usubjid descending roworder2;
run;

data relip2;
    set relip1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    relipflag=1;
run;

/*RELATED TO IP - EXPECTED*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=4 and
roworder2 in(5 6 7)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aeexpec aseq) out=expecip;
    by headorder1 headtext1 usubjid roworder2;
run;

```



```

proc sort data=expecip(keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aeexpec aeseq) nodupkey
out=expecip1;
    by headorder1 headtext1 usubjid roworder2;
run;

data expecip2;
    set expecip1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    expecipflag=1;
run;

/*RELATED TO NRT gum*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=5 and
roworder2 not in(1 5 6 7)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aere12n aere12 aeseq) out=relnrt;
    by headorder1 headtext1 usubjid descending roworder2;
run;

proc sort data=relnrt(keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aere12n aere12 aeseq) nodupkey
out=relnrt1;
    by headorder1 headtext1 usubjid descending roworder2;
run;

data relnrt2;
    set relnrt1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    relnrtflag=1;
run;

/*RELATED TO NRT gum - EXPECTED*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=5 and
roworder2 in(5 6 7)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aeexpec1 aeseq) out=expecnrt;
    by headorder1 headtext1 usubjid roworder2;
run;

proc sort data=expecnrt(keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aeexpec1 aeseq) nodupkey
out=expecnrt1;
    by headorder1 headtext1 usubjid roworder2;
run;

data expecnrt2;
    set expecnrt1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    expecnrtflag=1;
run;

```

```

/*RELATED TO SP*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=6 and
roworder2 not in(1 6 7)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aerelspn aerelsp aeseq) out=relsp;
    by headorder1 headtext1 usubjid descending roworder2;
run;

proc sort data=relsp(keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aerelspn aerelsp aeseq) nodupkey
out=relsp1;
    by headorder1 headtext1 usubjid descending roworder2;
run;

data relsp2;
    set relsp1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    relspflag=1;
run;

/*SEVERITIES*/
proc sort data=adverse04_a(where=(not missing(subjid) and roworder1=7 and
roworder2 not in(1)) keep=headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aebodsys aeterm aesev aesev aeseq) out=sev;
    by headorder1 headtext1 usubjid descending roworder2;
run;

proc sort data=sev(keep=headorder1 headtext1 roworder1 roworder2 rowtext
usubjid subjid aebodsys aeterm aesev aesev aeseq) nodupkey out=sev1;
    by headorder1 headtext1 usubjid descending roworder2;
run;

data sev2;
    set sev1;
    by headorder1 headtext1 usubjid;
    if first.usubjid;
    sevflag=1;
run;

/*MERGE BACK ONTO ORIGINAL DATA TO INCLUDE FLAGS*/

proc sort data=adverse04_a; by headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aeseq; run;
proc sort data=ser2; by headorder1 headtext1 roworder1 roworder2 rowtext
usubjid subjid aeseq; run;
proc sort data=relip2; by headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aeseq; run;
proc sort data=expecip2; by headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aeseq; run;
proc sort data=relnrt2; by headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aeseq; run;
proc sort data=expecnrt2; by headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aeseq; run;

```

```
proc sort data=relsp2; by headorder1 headtext1 roworder1 roworder2
rowtext usubjid subjid aeseq; run;
proc sort data=sev2; by headorder1 headtext1 roworder1 roworder2 rowtext
usubjid subjid aeseq; run;
```

```
data alldata;
    merge adverse04_a(in=a) ser2(in=b) relip2(in=c) relnrt2(in=d)
relsp2(in=e) sev2(in=f) expecip2(in=g) expecnrt2(in=h);
    by headorder1 headtext1 roworder1 roworder2 rowtext usubjid subjid
aeseq;
    if b and not a then serflag=1;
    if c and not a then relipflag=1;
    if d and not a then relnrtflag=1;
    if e and not a then relspflag=1;
    if f and not a then sevflag=1;
    if g then expecipflag=1;
    if h then expecnrtflag=1;
run;
```

```
proc sql;
    create table results02_ser as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(serflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;
```

```
proc sql;
    create table results02_relip as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(relipflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;
```

```
proc sql;
    create table results02_expecip as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(expecipflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;
```

```
proc sql;
    create table results02_relnrt as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(relnrtflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;
```

```
proc sql;
    create table results02_expecnrt as
```

```

    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(expecnrtflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;

proc sql;
    create table results02_relsp as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(relspflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;

proc sql;
    create table results02_sev as
    select headorder1, headtext1, roworder1, roworder2, rowtext, subjid,
count(distinct subjid) as subjects
    from alldata(where=(sevflag=1))
    group by headorder1, headtext1, roworder1, roworder2, rowtext;
quit;

proc sort data=results02_ser; by headorder1 headtext1 roworder1 roworder2
rowtext subjid; run;
proc sort data=results02_relip; by headorder1 headtext1 roworder1
roworder2 rowtext subjid; run;
proc sort data=results02_relnrt; by headorder1 headtext1 roworder1
roworder2 rowtext subjid; run;
proc sort data=results02_relsp; by headorder1 headtext1 roworder1
roworder2 rowtext subjid; run;
proc sort data=results02_sev; by headorder1 headtext1 roworder1 roworder2
rowtext subjid; run;
proc sort data=results02_expecip; by headorder1 headtext1 roworder1
roworder2 rowtext subjid; run;
proc sort data=results02_expecnrt; by headorder1 headtext1 roworder1
roworder2 rowtext subjid; run;

data allcounts;
    set results02_ser results02_relip results02_relnrt results02_relsp
results02_sev results02_expecip results02_expecnrt;
    by headorder1 headtext1 roworder1 roworder2 rowtext subjid;
    subjects1=subjects;
run;

proc sort data=results02; by headorder1 headtext1 roworder1 roworder2
rowtext subjid; run;
proc sort data=results02_allaes; by headorder1 headtext1 roworder1
roworder2 rowtext subjid; run;

data results02_comb;
    merge allcounts(in=a) results02(where=(roworder1 not in(2 8)) in=b)
results02_allaes(rename=(events=events1 subjects=subjects1)in=c);
    by headorder1 headtext1 roworder1 roworder2 rowtext subjid;
    if b and a then test=1;

```

```

run;

data results03;
  merge results02_comb(in=a) results01(keep=headorder1 headtext1
treated);
  by headorder1 headtext1;
  if a;

  subjects=subjects1;
  if missing(events) then events=events1;

  if events ne 0 and missing(subjid) then delete;

  drop subjects1 events1;
run;

data results04;
  set results01 results03;

  if index(rowtext,'Missing') and events=0 and (subjects=0 or
missing(subjects)) then delete;
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;

```

```

set results06;
length text text2 text3 $200.;
if (events = . and subjects = .) or missing(events) and
missing(subjects) then do;
    events = 0;
    subjects = 0;
end;

    if treated ne 0 then percent = 100 * subjects / treated;

if missing(roworder1) or roworder1 = 1 then delete;

    if roworder1 ne 2 and roworder2 = 1 then do;
        text='';
        text2='';
        text3='';
    end;
    else if roworder1 eq 8 and roworder2 = 4 then do; /*Blank row for
treatments given header*/
        text='';
        text2='';
        text3='';
    end;
    else do;
        /*n value*/
        if missing(subjects) then text='0';
        else text=put(subjects,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(events) OR EVENTS=0 then text2='';
        else text2=compress(put(events,3.));
    end;

    if roworder2 ne 1 and missing(subjects) and not missing(subjid)
then delete; /*This deletes subjects where it isn't worst case scenario*/
/* 6) JMH 24Mar2014 */
    keep headorder1 headtext1 roworder1 roworder2 rowtext text text2 text3;
run;

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

data dumtrts; /*Use this to output any columns for which N=0*/
    attrib headtext1 length =$200.

```

```

                                rowtext length=$70.
                                headorder1 length=8.;

roworder1=2;
roworder2=1;
rowtext='Adverse events (AE)';

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;

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;
    text='0';
    text2='';
    text3='';
  end;

  if roworder1=4 and roworder2=6 and left(strip(text))='0' and
left(strip(text2))='' then delflag=1;
  if roworder1=4 and roworder2=7 and left(strip(text))='0' and
left(strip(text2))='' then delflag=1;
  if roworder1=5 and roworder2=6 and left(strip(text))='0' and
left(strip(text2))='' then delflag=1;
  if roworder1=5 and roworder2=7 and left(strip(text))='0' and
left(strip(text2))='' then delflag=1;

run;

proc sort data=results07a; by roworder1 roworder2 rowtext delflag ; run;

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

```

```

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

proc transpose data=results07a out=results08_p prefix=p;
  by roworder1 roworder2 rowtext;
  id headorder1;
  idlabel headtext1;
  var text3 ;
run;

data results08;
  merge results08_n results08_e results08_p;
  by roworder1 roworder2;
run;

proc sort data=results07a out=delflags(keep=roworder1 roworder2 delflag)
nodupkey;
  by roworder1 roworder2;
run;

data results08a;
  merge results08 delflags;
  by roworder1 roworder2;
run;

proc transpose data=results08a(where=(roworder1 in (4 5) and roworder2
in(6 7))) out=tdelflags prefix=d;
  by roworder1;
  id roworder2;
  idlabel rowtext;
  var delflag;
run;

data tdelflags2;
  set tdelflags;
  if d6=1 and d7=1 then remove=1;
run;

proc transpose data=tdelflags2(where=(not missing(remove)))
out=removedata;
  by roworder1;
  id remove;
  var d6 d7;
run;

data removedata1;
  set removedata;
  rename _label_=rowtext;

```



```

        rename _1=remove;
        roworder2=input(left(strip(tranwrd(_name_,'D','')),8.));
        drop _name_;
run;

data results08b;
    merge results08 removedata1;
    by roworder1 roworder2 rowtext;
    if remove=1 then delete;
run;

data missflags;
    set results08b;
    where index(rowtext,'Related');
    if n1='0' or missing(n1) then mflag1=1;
    if n2='0' or missing(n2) then mflag2=1;
    if n3='0' or missing(n3) then mflag3=1;
    if n4='0' or missing(n4) then mflag4=1;
    if n6='0' or missing(n6) then mflag6=1;
    keep roworder1 mflag;;
run;

proc sort data=results08b; by roworder1; run;
proc sort data=missflags; by roworder1 ; run;

data results08c;
    merge results08b missflags;
    by roworder1;
run;

data labels;
    set results08c;
    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";

    if roworder2 not in (6 7) then do;

```

```

        mflag1=.;
        mflag2=.;
        mflag3=.;
        mflag4=.;
        mflag6=.;
    end;

run;

data final4;
    set labels;
        attrib rowtext_x length=$200. label='Unformatted text'
                rowtext label='Formatted text';

if roworder2 ne 1 and rowtext ne '$S={foreground=white} . $S={} Treatment
given' then do;
    array a [3] n1 n2 n4;
    array b [3] e1 e2 e4;
    array c [3] mflag1 mflag2 mflag4;
    do i=1 to 3;
        if c[i] ne 1 then do;
            if missing(a[i]) then a[i] ='0';
            end;
            else if c[i] eq 1 then do;
                if a[i]='0' then a[i]='';
                if b[i]='0' then b[i]='';
            end;
            if length(left(strip(b[i])))=1 then b[i]=
'$S={foreground=white}.$S={} ' || left(strip(b[i]));
        end;

        if missing(n99) then n99='0';
        if length(left(strip(e99)))=1 then e99=
'$S={foreground=white}.$S={} ' || left(strip(e99));
        end;

        flag=1;

        /*Create unformatted variable for qc*/

        rowtext_x=tranwrd(rowtext,'$S={foreground=white} .
$S={} ',' ');
        rowtext_x=tranwrd(rowtext_x,'$S={foreground=white}
$S={} ',' ');

        rowtext_x=tranwrd(rowtext_x,' $n ',' ');
        rowtext_x=tranwrd(rowtext_x,' ',' ');
        rowtext_x=left(trim(rowtext_x));

        if roworder1 in(2 3 4) then pageord=1;
        else if roworder1 in(5 6 7) then pageord=2;
        else if roworder1=8 then pageord=3;
        else put "WA" "RNING: Assign pageord for roworder1: "

roworder1= ;

```

```

run;

proc sql noprint;
    create table table.T_15_02_06_01 as
        select rowtext, rowtext_x, n1, p1, e1, n2, p2, e2, n3, p3, e3, n4,
        p4, e4, n6, e6, p6, n99, p99, e99
        from final4
        order by pageord, roworder1, roworder2;
quit;

data paging;
    set final4;
        by pageord roworder1 roworder2;
            if first.pageord/* LN>9*/ then ln=1; /*Amend to look
presentable, and avoid page overflows*/ /* 2) AOB 12Aug2014 */ /* 5)
AOB 12Aug2014 */
            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=130, 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 15.2.6.1 Summary of Adverse Events - 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 nowd split = '$' %if &i=1 %then
%do; contents=' ' %end; %else %do; contents='' %end;;;
column flag page pageord roworder1 roworder2 rowtext ("Sequence
&linebot" ("THS 2.2 Menthol$(N=&trt1) &linebot"/*((" n (%)
Events"*/ n1 p1 e1)/ * 1) AOB 12Aug2014 */ /* 5) AOB 12Aug2014 */

("mCC -$THS 2.2 Menthol$(N=&trt2) &linebot" /*(" n (%) Events"*/
n2 p2 e2) /* 1) AOB 12Aug2014 */ /* 5) AOB 12Aug2014 */

("THS 2.2 Menthol$(N=&trt3) &linebot" /*(" n (%)
Events"*/ n3 p3 e3) /* 1) AOB 12Aug2014 */ /* 5) AOB 12Aug2014 */

("NRT gum -$THS 2.2 Menthol$(N=&trt4) &linebot" /*(" n (%)
Events"*/ n4 p4 e4) /* 1) AOB 12Aug2014 */ /* 5) AOB 12Aug2014 */

("Exposed Not$Randomized$(N=&trt6) &linebot" /*(" n (%) Events"*/
n6 p6 e6)) /* 1) AOB 12Aug2014 */ /* 5) AOB 12Aug2014 */

("Overall$Safety$(N=&trt99) &linebot" /*(" n (%) Events"*/ n99 p99
e99); ; /* 1) AOB 12Aug2014 */ /* 5) AOB 12Aug2014 */
define flag / order order = internal noprint;
define page / order order = internal noprint;
define pageord / order order = internal noprint;
define roworder1 / order order = internal noprint;
define roworder2 / order order = internal noprint;

```

```

        define rowtext          / display style={just=left
cellwidth=3/*2.8*/cm}' '; /* START 1)  AOB 12Aug2014 */ /* START 5)  AOB
12Aug2014 */
        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};

/*      DEFINE N1-N4          / DISPLAY STYLE={JUST=D CELLWIDTH=0.2CM}"";*/
/*      DEFINE N6            / DISPLAY STYLE={JUST=D CELLWIDTH=0.2CM}"";*/
/*      DEFINE N99           / DISPLAY STYLE={JUST=D
CELLWIDTH=0.2CM}"";*/
/*      DEFINE E1-E4         / DISPLAY STYLE={JUST=C
CELLWIDTH=0.3CM}"";*/
/*      DEFINE E6            / DISPLAY STYLE={JUST=C
CELLWIDTH=0.3CM}"";*/
/*      DEFINE E99           / DISPLAY STYLE={JUST=C
CELLWIDTH=0.3CM}"";*/
/*      DEFINE P1            / DISPLAY STYLE={JUST=C
CELLWIDTH=1.1CM}"";*/
/*      DEFINE P2            / DISPLAY STYLE={JUST=C
CELLWIDTH=1.1CM}"";*/

```

```

/*          DEFINE P3          / DISPLAY STYLE={JUST=C
CELLWIDTH=1.1CM}"";*/
/*          DEFINE P4          / DISPLAY STYLE={JUST=C
CELLWIDTH=1.1CM}"";*/
/*          DEFINE P6          / DISPLAY STYLE={JUST=C
CELLWIDTH=1.1CM}"";*/
/*          DEFINE P99         / DISPLAY STYLE={JUST=C
CELLWIDTH=1.1CM}"";*/
/* END 1)  AOB 12Aug2014 */ /* END 5)  AOB 12Aug2014 */

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

break after page / page;

compute after roworder1;
    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: IP = Investigational Product (THS 2.2 Menthol /
mCC).';
    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_01.lst" new;
run;

proc contents data = table.T_15_02_06_01 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
*  END OF PROGRAM CODE                               ;
*=====;
```