

```

%_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_rsndis.sas;
%put NOTE: Purpose              : table of reasons for discontinuations;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADDE SDTM.DE;
%put NOTE: Output               : t_15_2_1_2(rod);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jhardman;
%put NOTE: Creation Date        : 2014-07-31;
%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_01_02(rod);

/* 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 ;
*****;

/*Read in ADSL for column headers*/
data adsl;
    set adam.adsl;
    where randfl='Y';
    trtan=trt01pn;
    trta=trt01p;
    output;
    trtan=99;
    trta='Overall Randomized';
    output;
run;

data dumtrts;
attrib trta length=$200. trtan length=8.;
    trtan=1;
    trta='THS 2.2';
    output;
    trtan=2;
    trta='CC';
    output;
    trtan=3;
    trta='SA';
    output;
run;

proc sort data=adsl;
    by trtan trta;
run;

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

data tot2;
    merge dumtrts(in=a) tot(in=b);
    if a or b;
    if a and not b then count=0;
    by trtan trta;
    rename count=total;
    call symput('trt' || compress(put(trtan,best.)), compress(count));
run;

/*Bring in ADDS to get subjects who did not complete the study*/

data adds;
    set adam.adds;
    where randfl = 'Y' and complfl='N';

```

```

        attrib trta1 length=$200.;
        trta1=trta;
        drop trta;
        rename trta1=trta;
        output;
        trtan=99;
        trta1='Overall Randomized';
        output;
run;

proc sort data=adds;
    by usubjid;
run;

data ds02;
    set adds;
    where dscat='DISPOSITION EVENT';
run;

proc sort data=ds02;
    by trtan trta;
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 discontinuations;
data ds03;
    set ds02;
    by trtan trta;
    output;

    if first.trta then do;
        subjid = .;
        dsdecod='';
        output;
    end;
run;

* Create values for table rows;
data ds04;
    set ds03;
    length rowtext $70;
    * All discontinuations;
    roworder1 = 2;
    roworder2 = 1;
    rowtext = 'Total no. of discontinuations - n (%)';
    output;
    * Reasons for Discontinuations - header;
    roworder1 = 3;
    roworder2 = 1;
    rowtext = 'Reason for discontinuation';

```

```

if subjid= . then output;
* Adverse events;
roworder1 = 3;
roworder2 = 2;
rowtext = '$S={foreground=white} . $S={} Adverse events - n (%)';
if subjid = . or dsdecod = 'ADVERSE EVENT' then output;
* Protocol violation;
roworder1 = 3;
roworder2 = 3;
rowtext = '$S={foreground=white} . $S={} Protocol violation - n (%)';
if subjid = . or dsdecod = 'PROTOCOL VIOLATION' then output;
    *Withdrawal by subject - header;
roworder1 = 3;
roworder2 = 4;
rowtext = '$S={foreground=white} . $S={} Withdrawal by subject - n
(%)';
if subjid = . or dsdecod = 'WITHDRAWAL BY SUBJECT' then output;
    *Lost to follow-up;
roworder1 = 3;
roworder2 = 5;
rowtext = '$S={foreground=white} . $S={} Lost to follow-up - n (%)';
if subjid = . or dsdecod = 'LOST TO FOLLOW-UP' then output;
    *Any other reason;
roworder1 = 3;
roworder2 = 6;
rowtext = '$S={foreground=white} . $S={} Other - n (%) ';
if subjid = . or dsdecod = 'OTHER' then output;

/*New section only needed if sfaety follow up is incomplete*/
if randfl='Y' and fupfl='N' then do;
    * All discontinuations;
roworder1 = 4;
roworder2 = 1;
rowtext = 'Total no. of incomplete safety follow-up - n (%)';
output;
    * Reasons for Discontinuations - header;
roworder1 = 5;
roworder2 = 1;
rowtext = 'Reason for discontinuation';
if subjid= . then output;
    * Adverse events;
roworder1 = 5;
roworder2 = 2;
rowtext = '$S={foreground=white} . $S={} Adverse events - n (%)';
if subjid = . or dsdecod = 'ADVERSE EVENT' then output;
    * Protocol violation;
roworder1 = 5;
roworder2 = 3;
rowtext = '$S={foreground=white} . $S={} Protocol violation - n (%)';
if subjid = . or dsdecod = 'PROTOCOL VIOLATION' then output;
    *Withdrawal by subject - header;
roworder1 = 5;
roworder2 = 4;

```

```

    rowtext = '$S={foreground=white} . $S={} Withdrawal by subject - n
(%)';
    if subjid = . or dsdecod = 'WITHDRAWAL BY SUBJECT' then output;
        *Lost to follow-up;
    roworder1 = 5;
    roworder2 = 5;
    rowtext = '$S={foreground=white} . $S={} Lost to follow-up - n (%)';
    if subjid = . or dsdecod = 'LOST TO FOLLOW-UP' then output;
        *Any other reason;
    roworder1 = 5;
    roworder2 = 6;
    rowtext = '$S={foreground=white} . $S={} Other - n (%) ';
    if subjid = . or dsdecod = 'OTHER' then output;
end;
run;

proc sort data=tot2;
    by trta trtan;
run;
proc sort data=adds;
    by trta trtan;
run;

data adds1;
    merge tot2 adds;
    by trta trtan;
run;

data results01;
    set adds1;
    treated=total;
    dsterm=dsdecod;
run;

data ds04_x;
    set ds04;
    dsterm=dsdecod;
run;

proc sort data=ds04_X out=ds04_a nodupkey;
    by trtan trta roworder1 roworder2 rowtext usubjid dsterm;
run;

proc sql;
    create table results02 as
    select trtan, trta, roworder1, roworder2, rowtext, usubjid,
count(dsterm) as events,
        count(distinct subjid) as subjects
    from ds04_a
    group by trtan, trta, roworder1, roworder2, rowtext;
quit;

proc sort data=results01 nodupkey out=results01_x;
    by trtan trta;

```

```

run;

data results03;
  merge results02(in=a) results01_x(keep=trtan trta treated);
  by trtan trta;
  if a;
run;

proc sort data=results01;
  by trtan trta;
run;

data results04;
  merge results01_x(in=a) results03(in=b);
  by trtan trta ;
  if a and not b then do;
    roworder1=2;
    roworder2=1;
    rowtext='Total no. of discontinuations - n (%)';
    events=0;
    subjects=0;
  end;
  if missing(trtan) then delete;
run;

proc sort data=results04;
  by trtan trta 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 trtan, trta, 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 trtan trta 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 trtan trta roworder1 roworder2 rowtext ;
run;

```

```

* Convert results to text values for the summary table;
data results07;
  set results06;
  length text $20 ;
  if (events = . and subjects = .) or missing(events) and
missing(subjects) then do;
    events = 0;
    subjects = 0;
  end;
  if total ne 0 then do;
    percent = 100 * subjects / total;
  end;
  else do percent=0;
  end;

      if percent ne 0 then text= put(subjects,2.) || ' ( ' ||
compress(put(percent,8.1)) || '%)';
      else text='';

run;

proc sort data=results07 nodupkey;
  by trtan trta roworder1 roworder2 rowtext text;
run;

/*Use this to output any columns for which N=0*/

data dumtrts1;
  set dumtrts;

  roworder1=2;
  roworder2=1;
  rowtext='Total no. of discontinuations - n (%)';
run;

data results07a;
  merge results07(in=a) dumtrts1(in=b);
  by trtan trta roworder1 roworder2 rowtext;
  if a or b;
  if rowtext='Total no. of discontinuations - n (%)' and
missing(text) then do;
    text='0          ';
  end;
run;

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

* Transpose the results;
proc transpose data=results07a out=results08_n prefix=n;
  by roworder1 roworder2 rowtext ;
  id trtan;

```

```

        idlabel trta;
    var text ;
run;

data results08;
    set results08_n;

    if roworder2 ne 1 then do;

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

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

    /*Creating sortord to break after each section*/
    if roworder1 in(2 3) then sortord=1;
    else sortord=2;

    flag=1;
run;

proc sql noprint;
    create table table.T_15_02_01_02 as
    select rowtext, n1, n2, n3, n99
    from results08
    order by sortord, roworder1, roworder2;
quit;

data paging;
    set results08;
    by sortord roworder1 roworder2 ;
    if first.sortord and ln ge 8 then ln=1; /*Amend to look
presentable, and avoid page overflows*/
    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.t106324 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106324/TFL/&TFL_Part./&tflno..rtf"
style=t106324 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 15.2.1.2 Summary of Reasons for
Discontinuations - Randomized 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 proclabel=' ';
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 = '$' %if &i=1 %then
%do; contents=' ' %end; %else %do; contents=' ' %end;;
    column flag page sortord roworder1 roworder2 rowtext ("THS
2.2$(N=&trt1)" N1) ("CC$(N=&trt2)" N2)

                ("SA$(N=&trt3)" N3) ("Overall$Randomized$(N=&trt99)"
N99);
    define flag          / order order = internal noprint;
    define page          / order order = internal noprint;
    define sortord       / order order = internal noprint;
    define roworder1     / order order = internal noprint;

```

```

        define roworder2      / order order = internal noprint;
        define rowtext        / display style={just=left
cellwidth=4cm}' ';
        define n1              / display style={just=left
cellwidth=1.1cm pretext="\tqdec\tx500 "} style(header)={just=center} "";
        define n2              / display style={just=center
cellwidth=1.1cm} style(header)={just=center} "";
        define n3              / display style={just=center
cellwidth=1.1cm} style(header)={just=center} "";
        define n99             / display style={just=left
cellwidth=1.2cm 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;

        compute before page / style={protectspecialchars=off};;
        line "&linetop";
        endcomp;

        compute after sortord;
        line "" ;
        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: Percentages are based on the number of subjects
indicated in the column header (N).';
        line ' ';
        line 'Appendix 15.3.1.7';
        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_02.lst" new;
run;

proc contents data = table.T_15_02_01_02 varnum;
run;
ods listing close;

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