

```

%_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_demog2.sas;
%put NOTE: Purpose              : table of demographics FAS;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADBX ADAM.ADSL ADAM.ADQSD;
%put NOTE: Output               : t_15_2_1_4_2(dm);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jhardman;
%put NOTE: Creation Date        : 2014-07-29;
%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: 01Aug2014  JMH        1) Amended order and column header;
%put NOTE: 01Sep2014  JMH        2) Added footnote as per cleint
comments;
%put NOTE: 16Sep2014  JR         3) FTND classification sorting
amended;
%put NOTE: 16Sep2014  JR         4) Amended dual programming sorting;
%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_01_04_02(dm);

%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 allpk;
  set adam.adsl(where=(fasfl='Y' and trt01an ne 98 and trt01an ne 97));
  format trta $40.;

  trta='Overall Safety';
  trtan=9;
  output;
run;

/*Find the number of people in each treatment sequence for table
headers*/
data adsl;
  set adam.adsl(where=(fasfl='Y' and trt01an ne 98 and trt01an ne 97));
  format trta $40.;

  trta=trt01a;
  trtan=trt01an;
run;

data adsl1;
  set adsl allpk;
run;

proc sort data=adsl1;
  by trta trtan;
run;

proc freq data=adsl1 noprint;
  tables trta*trtan / out=adsltotst(where=(not missing(trtan)))
drop=percent);
run;

data dummy;
  format trta $40.;
  trta='THS 2.2';
  trtan=1;
  output;
  trta='CC';
  trtan=2;
  output;
  trta='SA';
  trtan=3;
  output;
  trta='Overall Safety';
  trtan=9;
  output;
run;

```

```

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

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

data adsltot2;
  merge adsltot dummy;
  by trtan trta;

  if count=. then count=0;
run;

data tot2a;
  set adsltot2;

  rename count=total;

  call symput('trt'||strip(put(trtan,best.)),strip(put(count,best.)));
run;

proc sort data=tot2a;
  by trtan trta;
run;

/*CYP and FTNS*/
/* Creating period 1 treatments */
data adsl2;
  set adam.adsl(where=(fasfl='Y' and trt01an ne 98 and trt01an ne
97));
  format trta $40.;

  trtan=trt01an;
  trta=trt01a;

  keep usubjid trtan trta;
run;

/* Obtaining data for CYP and FTND */
data cyp;
  set adam.adbx(where=(paramcd='CYP2A6' and fasfl='Y' and trtan ne 98
and trtan ne 97 and avisit='Day 0'));
  rename aval=cypn avalc=cypc;
  keep usubjid aval avalc;
run;

proc sort data=adam.adqsnd(where=(paramcd='FTNDSC' and fasfl='Y' and
trtan ne 98 and trtan ne 97)) out=adqsnd(keep=usubjid avalcat1 aval
rename=(avalcat1=ftcat aval=ftsc));
  by usubjid;
run;

```

```

data cypqs;
    merge adsl2 adqsnd cyp;
    by usubjid;
    if ftcats='Mild' then ftcatsn=1;
    else if ftcats='Moderate' then ftcatsn=2;
    else if ftcats='Severe' then ftcatsn=3;
    else if missing(ftcats) then ftcatsn=4;

run;

data cypqs2;
    set cypqs;

run;

/* Overall PK data */
data cyppk;
    set adam.adbx(where=(paramcd='CYP2A6' and fasfl='Y' and trtan ne 98
and trtan ne 97 and avisit='Day 0')));

    trta='Overall Safety';
    trtan=9;
    output;
    rename aval=cypn avalc=cypc;
    keep usubjid aval avalc;
run;

data adqsndpk;
    set adam.adqsnd(where=(paramcd='FTNDSC' and fasfl='Y' and trtan ne 98
and trtan ne 97));
    trta='Overall Safety';
    trtan=9;
    output;
    rename avalcats=ftcats aval=ftsc;
    keep usubjid avalcats aval ;
run;

data cypqspk;
    merge allpk(keep=usubjid trtan trta) adqsndpk cyppk;
    by usubjid;
    if ftcats='Mild' then ftcatsn=1;
    else if ftcats='Moderate' then ftcatsn=2;
    else if ftcats='Severe' then ftcatsn=3;
    else if missing(ftcats) then ftcatsn=4;

run;

data cypqsall;
    set cypqs2 cypqspk;

run;

proc sort data=cypqsall;
    by usubjid trtan trta;
run;

```

```

proc sort data=adsl1;
  by usubjid trtan trta;
run;

data adslall;
  merge adsl1 cypqsall;
  by usubjid trtan trta;
run;

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

/*Macro for all the class variables*/

%macro _class(var1=, var2=, dout1=, dout2=, var=, num=);

proc freq data=adslall noprint;
  table trtan*trta*&var1*&var2 / out =&dout1(drop=percent);
run;

data &dout2._1;
  merge &dout1 tot2a(where=(total ne 0));
  by trtan trta;
  percent=count/total*100; /*This works out the percentages*/
run;

data &dout2;
  set &dout2._1;
  format var $200. stat $20. svar $20.;
  order=&num;
  var=trim(&var2);
  stat='n (%)';
  if percent=100 then svar = strip(put(count,best.)) || ' ' || '('
||strip(put(percent,5.)) || ' %)';
  if 10<=percent<100 then svar = strip(put(count,best.)) || ' ' ||
'(' ' ||strip(put(round(percent,0.1),5.1)) || ' %)';
  if percent<10 then svar = strip(put(count,best.)) || ' ' || '(' '
||strip(put(round(percent,0.1),5.1)) || ' %)';
  run;

  proc sort data=&dout2 ; by order &var1 var stat; run;

  proc transpose data=&dout2 out=&dout2._2 prefix=t;
    by order &var1 var stat;
    var svar;
    id trtan;
  run;

/*Dummy will add a blank line, only the variable name for TFLs will be
presented*/
data dummy;
  format var $200. stat $20.;
  var="&var";

```

```

        stat=' ';
        order=&num;
    %if &num=5 %then %do;
        output;
        var='Underweight';
        stat='n (%)';
        output;
        var='Normal weight';
        stat='n (%)';
        output;
        var='Overweight';
        stat='n (%)';
        output;
        var='Obese';
        stat='n (%)';
        output;
    %end;
    %if &num =8 %then %do;
        output;

        var='WHITE';
        stat='n (%)';
        output;
        var='BLACK OR AFRICAN AMERICAN';
        stat='n (%)';
        output;
        var='AMERICAN INDIAN OR ALASKA NATIVE';
        stat='n (%)';
        output;
        var='ASIAN';
        stat='n (%)';
        output;
        var='NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER';
        stat='n (%)';
        output;
        var='OTHER';
        stat='n (%)';
        output;
    %end;
    %if &num=14 %then %do;
        output;
        var='Mild';
        stat='n (%)';
        output;
        var='Moderate';
        stat='n (%)';
        output;
        var='Severe';
        stat='n (%)';
        output;
        var='';
        stat='n (%)';
        output;
    %end;
run;

```

```

%if &num=8 | &num=5 | &num=14 %then %do;
  proc sort data=&dout2._2;
    by var;
  run;

  proc sort data=dummy;
    by var;
  run;
%end;

/*Combine the dummy and the datasets with actual data*/
data all&num.;
  format var $200. stat $20.;
  %if &num=8 or &num=5 or &num=14 %then %do; merge dummy
&dout2._2; %end; %else %do; set dummy &dout2._2; %end;
  %if &num=8 or &num=5 or &num=14 %then %do;
    by var;
  %end;
  %IF &NUM=1 %THEN %DO; /*Start 4) JR 16Sep2014 */
    IF VAR='Male' THEN SORT=1;
    ELSE IF VAR='Female' THEN SORT=2;
  %END;
  %IF &NUM=2 %THEN %DO;
    IF VAR='10-19 cig/day' THEN SORT=1;
    ELSE IF VAR='>19 cig/day' THEN SORT=2;
  %END;
  %IF &NUM=9 %THEN %DO;
    IF VAR='<= 0.6 mg' THEN SORT=1;
    ELSE IF VAR='> 0.6 - 1.0 mg' THEN SORT=2;
  %END;
  %IF &NUM=12 %THEN %DO;
    IF VAR='1-5 mg' THEN SORT=1;
    ELSE IF VAR='6-8 mg' THEN SORT=2;
    ELSE IF VAR='9-10 mg' THEN SORT=3;
  %END; /* end 4) JR 16Sep2014 */
  %if &num=8 %then %do;
    if var='WHITE' then sort=1;
    else if var='BLACK OR AFRICAN AMERICAN' then sort=2;
    else if var='AMERICAN INDIAN OR ALASKA NATIVE' then sort=3;
    else if var='ASIAN' then sort=4;
    else if var='NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER' then
sort=5;
    else if var='OTHER' then sort=6;
  %end;
  %if &num=5 %then %do;
    if var='Underweight' then sort=1;
    else if var='Normal weight' then sort=2;
    else if var='Overweight' then sort=3;
    else if var='Obese' then sort=4;
  %end;
  %if &num=14 %then %do;
    if var='Mild' then sort=1;
    else if var='Moderate' then sort=2;

```

```

        else if var='Severe' then sort=3;

        if var='' then var='Missing';
        if var='Missing' then sort=/*4*/0.5; /* 3) JR 16Sep2014 */
    %end;
    drop &var1;
run;

%if &num=8 | &num=14 | &num=5 %then %do;
    proc sort data=all&num.;
        by sort;
    run;
%end;

%mend _class;
%_class(var1=sexn, var2=sexc, dout1=sex, dout2=sex2, var=Sex, num=1);
%_class(var1=bmigr1n, var2=bmigr1, dout1=bmiclass, dout2=bmiclass2,
var=BMI classifications,num=5);
%_class(var1=racen, var2=race, dout1=race, dout2=race2, var=Race, num=8);
%_class(var1=ucpdgr1n, var2=ucpdgr1, dout1=ccconsum, dout2=ccconsum2,
var=Daily CC consumption at screening,num=2);
%_class(var1=ftcatn, var2=ftcat, dout1=ftsc1, dout2=ftsc2, var=FTND total
score classifications,num=14);
%_class(var1=targr1n, var2=targr1, dout1=targr1, dout2=targr2, var=CC ISO
tar yield at admission,num=12/*11*/); /* 1) JMH 01Aug2014 */
%_class(var1=nicogr1n, var2=nicogr1, dout1=nicogr1, dout2=nicogr2,
var=ISO nicotine level,num=9);

/*End of macro for all the class variables*/

data adslall1;
    set adslall;

    height2=height/100;
run;

proc means data=adslall1 noprint;
    by trtan trta;
    var age height2 weightb1 bmi nicobl ftsc cypn tarb1; /*Make sure all
variables in the split macro are here too*/
    output out=stats;
    output out=median median=;
run;

data stats1;
    set stats median(in=in2);
    if in2 then _stat_ = 'MEDIAN';
    rename _stat_=_name_;
run;

data stats2;
    set stats1;

```



```

format stat $20.;
if _name_='N'      then order2=1001;
if _name_='MEAN'   then order2=1002;
if _name_='STD'    then order2=1003;
if _name_='MEDIAN' then order2=1004;
if _name_='MIN'    then order2=1005;
if _name_='MAX'    then order2=1006;

if _name_ eq 'STD' then stat='SD';
else if _name_ eq 'N' then stat='n';
else stat=propcase(_name_);

run;

%macro split(set=,num=,var=,stat=,units=,dp=);

data &set;
    format stat $20. svar $20.;
    set stats2(keep=trtan trta order2 stat &var);

    order=&num;

    %if &num=3 or &num=15 or &num=11/*12*/ %then %do; /* 1) JMH
01Aug2014 */
        if order2=1003 then
svar=compress(put(0.01*ceil(&var/0.01),8.2));
        else if order2=1001 then svar=compress(put(&var,8.));
        else if order2=1002 or order2=1004 then
svar=compress(put(&var,8.%eval(&dp+1)));
        else svar=compress(put(&var,8.&dp));
    %end;
    %if &num=4 or &num=10 or &num=7 %then %do;
        if order2=1003 then
svar=compress(put(0.001*ceil(&var/0.001),8.3));
        else if order2=1001 then svar=compress(put(&var,8.));
        else if order2=1002 or order2=1004 then
svar=compress(put(&var,8.%eval(&dp+1)));
        else svar=compress(put(&var,8.&dp));
    %end;
    %if &num=13 | &num=6 %then %do;
        if order2=1003 then
svar=compress(put(0.0001*ceil(&var/0.0001),8.4));
        else if order2=1001 then svar=compress(put(&var,8.));
        else if order2=1002 or order2=1004 then
svar=compress(put(&var,8.%eval(&dp+1)));
        else svar=compress(put(&var,8.&dp));
    %end;

run;

/*Combine meand and SD*/
data meansd&num;

```

```

        set &set(keep=trtan trta stat order2 svar &var);
        where stat in('Mean' 'SD'); /*Only keep mean and SD as this
is all we want here*/
        run;

        proc sort data=meansd&num; by trtan trta; run;

        proc transpose data=meansd&num out=meansd&num.a prefix=m;
            by trtan trta;
            id order2;
            var svar;
        run;

        data meansd&num.b;
            format stat $20. svar $20.;
            set meansd&num.a;
            svar=left(compress(m1002)) ||' (' || left(compress(m1003)) ||
')';

            stat='Mean (SD)';
            order2=1002;
            order=&num;
            drop m1002 m1003;
        run;
        /*End of combining mean and SD*/

```

```

/*Combine Min and Max*/
        data minmax&num;
            set &set(keep=trtan trta stat order2 svar &var);
            where stat in('Min' 'Max'); /*Only keep min and max as this
is all we want here*/
        run;

```

```

        proc sort data=minmax&num; by trtan trta; run;

        proc transpose data=minmax&num out=minmax&num.a prefix=m;
            by trtan trta;
            id order2;
            var svar;
        run;

```

```

        data minmax&num.b;
            format stat $20. svar $20.;
            set minmax&num.a;
            svar=left(compress(m1005)) ||', ' || left(compress(m1006));
            stat='Min, Max';
            order2=1005;
            order=&num;
            drop m1005 m1006;
        run;

```

```

/*End of combining min and max*/

```

```

/*Combine the new datasets which have Min, Max and Mean (SD)*/
        data minmaxmeansd;

```

```

        set minmax&num.b meansd&num.b;
run;

/*Add these new datasets to the original dataset, deleting individual
min, max, mean and SD*/
    data comb&set;
        set &set(where=(order2 not in(1002 1003 1005 1006)))
minmaxmeansd;
run;

proc sort data= comb&set; by order order2 stat; run;

/*Transpose the data by treatment sequence*/
proc transpose data=comb&set out=trans&set prefix=t;
    by order order2 stat;
    var svar;
    id trtan;
run;

/*This dummy will add in line with only the variable name to present in
the TFLs*/
    data dummy;
        format var $200. stat $20.;
        var="&stat (&units)";
        stat="n";
        order=&num;
        order2=1001;
run;

proc sort data=dummy; by order order2 stat; run;
proc sort data=trans&set; by order order2 stat; run;

/*Add the dummy to the transposed dataset*/
data all&num.;
    format var $200. stat $20.;
    merge dummy trans&set;
    by order order2 stat;
run;

proc sort data=all&num.; by order2; run;

%mend split;

%split(set=age,      num=3, var=age,      stat=Age,      units=years,
dp=0);
%split(set=bmi,      num=4, var=bmi,      stat=BMI,
units=kg/m{\super 2}, dp=1);
%split(set=nicnum,   num=10, var=nicobl, stat=CC nicotine yield at
admission, units=mg, dp=1);
%split(set=ftsc,     num=15, var=ftsc,     stat=FTND total score,
dp=0);

```

```

%split(set=cyp,      num=13, var=cypn,      stat=CYP2A6 activity,
units=%, dp=2);
%split(set=height,   num=6,  var=height2, stat=Height, units=m, dp=2);
%split(set=weight,   num=7,  var=weightbl, stat=Weight, units=kg, dp=1);
%split(set=tarbl,    num=11/*12*/, var=tarbl, stat=CC ISO tar yield at
admission, units=mg, dp=0); /* 1) JMH 01Aug2014 */

```

```

/*End of macro for continuous variables*/

```

```

/*Combine all the dataets which have been made from the macros and
make any formatting changes*/

```

```

data combined;
set all1 all2 all3 all4 all5 all6 all7 all8 all9 all10 all11
all12 all13 all14 all15;

```

```

if order=8 then var=propcase(var);
if order=8 then var=tranwrd(var,'Or','or');
if order=15 and index(var,'()') then var=tranwrd(var,'(',')');

attrib wrap length = $200;
wrap = var;

if order=5 then do;
i=37; *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 and j gt 1);
if substr(wrap,j,1)=' ' then do;
wrap=substr(wrap,1,j-1) || "$n $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;
end;
if order=8 then do;
i=33; *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 and j gt 1);
            if substr(wrap,j,1)=' ' then do;
                wrap=substr(wrap,1,j-1) || "$n $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;
end;

var=wrap;

```

```

        if order in (1 2 5 8 9 /*11*/12 14) and not missing(stat) then
var='$S={foreground=white} . $S={} ' || var; /* 1) JMH 01Aug2014 */

```

```

        if order in (1 2 3 4) then pageord=1;
        else if order in (5 6 7) then pageord=2;
        else if order in (8 9) then pageord=3;
        else if order in (10 11 12) then pageord=4;
        else if order in (13 14 15) then pageord=5;

```

```

        if order in (1 2 5 8 14) then do;
            array prod [4] t1 t2 t3 t9;
            do i=1 to 4;
                if not missing(stat) and missing(prod[i]) then prod[i]='0';
            end;
        end;

```

```

        run;

```

```

proc sort data=combined;
    by pageord order ORDER2 SORT; /* 4) JR 16Sep2014 */
run;

```

```

proc sql noprint;
    create table table.t_15_02_01_04_02 as
    select var, stat, t1, t2, t3, t9
    from combined
    order by PAGEORD, order, order2, SORT; /* 4) JR 16Sep2014 */
quit;

```

```

data paging;
    set combined;
    by pageord order ORDER2 SORT; /* 4) JR 16Sep2014 */

```

```

        if first.pageord or ln gt 13 then ln=1; /*Check for page overflows,
this may need changing*/
        else ln+1;
        if ln=1 then page+1;
        call symput("page",compress(put(page,best.)));
        flag=1;
run;

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.tl06324 (read) ;
ods results off;
ods rtf toc_data/* contents*/
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;
ods proclabel = ' ';

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

    /* Amend title as needed */
    _firtitl="Table 15.2.1.4.2 Summary of Demographics and Other
Baseline Characteristics - FAS 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;

```

```

* 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;
ods listing close;
proc report data = comp headline headskip nowd split = '#' /*ps = 60 ls =
120*/%if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
        column flag page order
("Variable" var) ("Statistic" stat) ("THS 2.2#(N=&trt1)" t1 )
("CC#(N=&trt2)" t2 )
("SA#(N=&trt3)" t3 ) /*("Overall#Safety#(N=&trt9)" t9)*/
("Overall#(N=&trt9)" t9);      /* 1) JMH 01Aug2014 */

        define flag          / order order = internal noprint;
        define page          / order order = internal noprint;
        define order         / order order = internal noprint;
        define var           / display style={just=1 cellwidth=4.5cm}
""style(header)={just=center};
        define stat          / display style={just=1 cellwidth=1.5cm} ""
style(header)={just=center};
        define t1            / display style={just=c cellwidth=2.2cm} "";
        define t2            / display style={just=c cellwidth=2.2cm} "";
        define t3            / display style={just=c cellwidth=2.2cm} "";
        define t9            / display style={just=c cellwidth=2.2cm} "";

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

        break after page / page;

        compute after order;
                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: BMI = Body Mass Index.";

```

```

        LINE "Note: FTND = Fagerstrom Test for Nicotine Dependence.";
/* 2) JMH 01Sep2014 */
        line 'Note: Percentages are based on the number of subjects
indicated in the column header (N).';
        line ' ';
        line "Appendix 15.3.1.2, 15.3.1.6, 15.3.1.9, 15.3.6.15";
        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=67, halfblnk=N);
ods listing;
proc printto print = "&table./t_15_02_01_04_02.lst" new;
run;

proc contents data = table.t_15_02_01_04_02 varnum;
run;
ods listing close;

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

```