

```

%_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_sdisp.sas;
%put NOTE: Purpose              : table of subject disposition;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADSL ADAM.ADDX ADAM.ADFA;
%put NOTE: Output               : t_15_2_1_1(sd);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jriley;
%put NOTE: Creation Date        : 2014-08-05;
%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: 11Aug2014   JR        1) Amended zeros in output;
%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_01(sd);

/* 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;
run;
data addx;
    set adam.addx;
run;
data adfa;
    set adam.adfa;
run;
data blanks;
    set adsl(firstobs=1 obs=14);
    statusn=_n_;
    keep statusn;
run;

data testdata;
    set adam.adsl;
    if not missing(ptestdtm) or not missing(dtestdtm) then testfl='Y';
run;

proc freq data=testdata;
    table siteid /noprint out=scrn (rename=(count=ov_scrn));
    table scrffl / noprint
out=sfl(where=(scrffl='Y') rename=(count=ov_scrn));
    table scrffl*testfl /noprint out=noprod(where=(scrffl='Y' and
(missing(testfl))) rename=(count=ov_scrn));
    table scrffl*testfl / noprint out=prod(where=(scrffl='Y' and (not
missing(testfl))) rename=(count=ov_scrn));
run;

data adfa1;
    merge adfa testdata(keep=usubjid testfl);
    by usubjid;
run;

proc freq data=ADFA1(where=(scrffl='Y' and paramcd
in('WILLABL','NRWILLAB') and aval=2));
    table parcat1*testfl /noprint out=willths(where=(parcat1='THS 2.2'
and testfl='Y') rename=(count=ov_scrn));
    table parcat1*parcat2*testfl /noprint
out=willnns(where=(parcat1='NICOTINE REPLACEMENT THERAPY' and testfl='Y')
rename=(count=ov_scrn));
run;
proc freq data=adsl;
    tables trtseqan*enrlfl / noprint
out=enr(where=(enrlfl='Y') rename=(count=enr));
    tables trtseqan*enrlfl*enfl / noprint out=enr_nr(where=(enrlfl='Y'
and enfl='Y') rename=(count=enr_nr));

```

```

        tables trtseqan*enrlfl*enfl / noprint out=enr_r (where=(enrlfl='Y'
and enfl='N') rename=(count=enr_r));
        tables trtseqan*randfl*complfl / noprint
out=rand_c (where=(randfl='Y' and complfl='Y') rename=(count=rand_c));
run;

proc sort data=adam.adsl (where=(randfl='Y' and complfl='N'))
out=discon (keep=complfl trtseqan usubjid); by usubjid; run;
proc sort data=addx (where=(complfl='N' and avisitn=101))
out=discon_w1 (keep=complfl trtseqan usubjid); by usubjid; run;

proc sort data=adam.adex (where=(complfl='N' and avisitn=101))
out=adex (keep=complfl trtseqan usubjid); by usubjid; run;

data discon_w;
    set adex discon_w1;
run;

data discon_wo;
    merge discon (in=a keep=complfl trtseqan usubjid) discon_w (in=b
keep=complfl trtseqan usubjid) ;
    by usubjid;
    if a and not b;
run;

proc freq data=discon; table trtseqan*complfl /noprint
out=rand_d (rename=(count=rand_d)); run;
proc freq data=discon_w; table trtseqan*complfl /noprint
out=rand_dw (rename=(count=rand_dw)); run;
proc freq data=discon_wo; table trtseqan*complfl /noprint
out=rand_dwo (rename=(count=rand_dwo)); run;

data adsl2;
    set adsl (where=(complfl='N'));
    if trt01a in ('THS 2.2', 'NRT gum') or trt02a in ('THS 2.2', 'NRT gum')
then output;
run;

data adsl3;
    set adsl (where=(complfl='N' and trt01a not in ('THS 2.2', 'NRT gum')
and trt02a not in ('THS 2.2', 'NRT gum')));
run;
proc freq data=adsl3;
    tables siteid / noprint out=d_nru (rename=(count=ov_scrn));
run;

data trtseq;
    merge enr enr_nr enr_r rand_c rand_d rand_dw rand_dwo;
    by trtseqan;
    rand=enr_r;
    keep trtseqan enr enr_nr enr_r rand rand_c rand_d rand_dw rand_dwo;
run;

```

```

proc transpose data=trtseq out=trtseq2(drop= _label_);
    id trtseqan;
    var enr enr_nr enr_r rand rand_c rand_d rand_dw rand_dwo ;
run;

data trtseq3;
    set trtseq2;
    if missing(_1) then do;
        _1=0; missingone='Y';
    end;
    if missing(_2) then do;
        _2=0; missingtwo='Y';
    end;
    if missing(_3) then do;
        _3=0; missingthree='Y';
    end;
    if missing(_4) then do;
        _4=0;missingfour='Y';
    end;
    IF MISSING(_5) THEN DO;
        _5=0;MISSINGFIVE='Y';
    end;
    ov_scrn=_1 + _2 + _3 + _4 + _5;
    if missingone='Y' then _1=.;
    if missingtwo='Y' then _2=.;
    if missingthree='Y' then _3=.;
    if missingfour='Y' then _4=.;
    if missingfive='Y' then _5=.;
    statusn=_n+6;
    drop missingone missingtwo missingthree missingfour missingfive;
run;

data overall1;
    set scrn sfl noprod prod willlths willnns;
    statusn=_n_;
run;

data overall2;
    merge overall1 blanks ;
    by statusn;
run;

data overall2a;
    merge overall2 blanks;
    by statusn;
run;

data overall3;
    merge overall2a trtseq3;
    by statusn;
    length status $132.;
    sort=1;
    statusn=_n_;
    if ov_scrn=. then ov_scrn=0;
    if statusn=1 then status='Total screened - n';

```

```

        IF STATUSN=2 THEN STATUS='    Screen failures - n (%)';
        if statusn=3 then status='    Screening failures, without product
test - n (%)';
        if statusn=4 then status='    Screening failures, with product test
- n (%)';
        if statusn=5 then status='    Unwilling to use THS 2.2';
        if statusn=6 then status='    Unwilling to use NRT gum';
        if statusn=7 then status='    Enrolled - n (%)';
        if statusn=8 then status='    Enrolled, not randomized - n (%)';
        if statusn=9 then status='    Randomized - n (%)';
        if statusn=10 then status='Total randomized - n';
        if statusn=11 then status='    Completed - n (%)';
        if statusn=12 then status='    Discontinued - n (%)';
        if statusn=13 then status='    Discontinued, with randomized product
use';
        if statusn=14 then status='    Discontinued, without randomized
product use';
        if statusn in(7,8) then
            do;
                _1=.;
                _2=.;
                _3=.;
                _4=.;
                _5=.;
            end;
        bynum=1;
        keep statusn status _1 _2 _3 _4 _5 ov_scrn sort bynum;
run;

data adsl;
    set adam.adsl(where=(not missing(trtseqan)));
    output;
    trtseqan=99;
    trtseqa='OVERALL RANDOMIZED';
    output;
run;

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

data dumtrts;
    attrib trtseqa length=$200. trtseqan length=8.;
    dumres=0;
    rown=0.5;
    trtseqan=1;
    trtseqa='THS 2.2 Menthol - mCC';
    output;
    trtseqan=2;
    trtseqa='mCC - THS 2.2 Menthol';
    output;
    trtseqan=3;
    trtseqa='THS 2.2 Menthol - NRT gum';
    output;

```

```

        trtseqan=4;
        trtsega='NRT gum - THS 2.2 Menthol';
        output;
        trtseqan=5;
        trtsega='EXPOSED NOT RANDOMIZED';
        output;
        trtseqan=99;
        trtsega='OVERALL SAFETY';
        output;
run;

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

proc transpose data=tot2 out=sum prefix=_;
    id trtseqan;
    var total;
run;

data sum_a;
    set sum;
    bynum=1;
run;

data sum01;
    merge sum_a overall3(where=(statusn=1)keep=ov_scrn statusn bynum);
    by bynum;
    total_overall=ov_scrn;

    OV RAND=_1+_2+_3+_4;

    rename _1=total_1 _2=total_2 _3=total_3 _4=total_4 _5=total_5;
    drop _name_ _label_;
run;

proc sort data=overall3;
    by statusn;
run;

data all;
    merge sum01 overall3;
    by bynum;
    retain total_1 total_2 total_3 total_4;
    if statusn in(1,10) then do;
        p_ov=put(ov_scrn,best.);
        ov_scrn=.;
    end;

```

```

else if statusn not in(1,10) then do;
    if _1>0 then p1a=( _1/total_1)*100;
    if _2>0 then p2a=( _2/total_2)*100;
    if _3>0 then p3a=( _3/total_3)*100;
    if _4>0 then p4a=( _4/total_4)*100;
    if _5>0 then p5a=( _5/total_5)*100;
    if statusn not in (11, 12, 13,14) then do;
        if ov_scrn>0 then p_ova=(ov_scrn/total_overall)*100;
    end;
    else do;
        p_ova=(ov_scrn/ov_rand)*100;
    end;

    if p1a=100 then p1=trim('(|compress(put(p1a,8.))||' %')');
    else if not missing(p1a) and p1a>10 and p1a<100 then p1=trim('(|compress(put(p1a,8.1))||'%)');
    else if not missing(p1a) and p1a<10 then p1=trim('(|compress(put(p1a,8.1))||'%)');

    if p2a=100 then p2=trim('(|compress(put(p2a,8.))||' %')');
    else if not missing(p2a) and p2a>10 and p2a<100 then p2=trim('(|compress(put(p2a,8.1))||'%)');
    else if not missing(p2a) and p2a<10 then p2=trim('(|compress(put(p2a,8.1))||'%)');

    if p3a=100 then p3=trim('(|compress(put(p3a,8.))||' %')');
    else if not missing(p3a) and p3a>10 and p3a<100 then p3=trim('(|compress(put(p3a,8.1))||'%)');
    else if not missing(p3a) and p3a<10 then p3=trim('(|compress(put(p3a,8.1))||'%)');

    if p4a=100 then p4=trim('(|compress(put(p4a,8.))||' %')');
    else if not missing(p4a) and p4a>10 and p4a<100 then p4=trim('(|compress(put(p4a,8.1))||'%)');
    else if not missing(p4a) and p4a<10 then p4=trim('(|compress(put(p4a,8.1))||'%)');

    if p_ova=100 then p_ov=trim('(|compress(put(p_ova,8.))||' %')');
    else if not missing(p_ova) and p_ova>10 and p_ova<100 then
p_ov=trim('(|compress(put(p_ova,8.1))||'%)');
    else if not missing(p_ova) and p_ova<10 then p_ov=trim('(|compress(put(p_ova,8.1))||'%)');
    end;
    if statusn in(13,14) and ov_scrn=0 then delete;

if statusn=12 then do;
    if missing(_1) then _1=0;
    if missing(_2) then _2=0;
    if missing(_3) then _3=0;
    if missing(_4) then _4=0;
end;

if statusn in(13,14) then do;
    if missing(_1) then _1=0;

```

```

/*          if missing(_2) then _2=0;*/ /* 1) JR 11Aug2014 */
end;

        rename _1=_tot1 _2=_tot2 _3=_tot3 _4=_tot4 _5=_tot5
ov_scrn=totooverall;
run;

proc sort data=all;
        by statusn;
run;

data all2;
        set all;
        attrib wrap length = $200;
        wrap = status;
i=55; *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;

        if statusn not in(1,10) then status=wrap;

        if missing(totooverall) then do;
                totooverall = p_ov;
                p_ov = '';
        end;

        flag=1;
run;

proc transpose data=all2(where=(statusn in(4 5 6))) out=subfl;
        var totooverall;
        id statusn;
run;

```



```

data _null_;
    set subfl;
    if sum(_4,_5,_6)=0 then call symput("SUBFL",1);
run;

data all2_extra;
    set all2;
    if &subfl=1 then do;
        if statusn in(5 6) then delete;
    end;
run;

proc sql noprint;
    create table table.T_15_02_01_01 as
    select status, _tot1, p1, _tot2, p2, _tot3, p3, _tot4, p4,
    totoverall, p_ov
    from all2_extra
    order by statusn;
quit;

data paging;
    set all2_extra;
    by statusn;
    if ln gt 10 then ln=1;
    else ln+1;
    if ln=1 then page+1;
    call symput("page",compress(put(page,best.)));
run;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&P_PGSize 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;

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

    /* Amend title as needed */
    _firtitl="Table 15.2.1.1 Summary of Subject Disposition -
All Screened Subjects";
    _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 = '$' %if &i=1 %then
%do; contents=' ' %end; %else %do; contents='' %end;;
    column flag page statusn ("Status$"status) ("Sequence &linebot"
("THS 2.2 Menthol$- mCC $(N=&trt1)" _tot1 P1)
    ("mCC -$THS 2.2 Menthol$(N=&trt2)" _tot2 p2) ("THS 2.2 Menthol -
$NRT gum $(N=&trt3)" _tot3 P3)
    ("NRT gum -$THS 2.2 Menthol$(N=&trt4)" _tot4 p4))
    ("Overall$Screened$" totoverall P_OV);
    define flag          / order order = internal noprint;
    define page          / order order = internal noprint;
    define statusn       / order order = internal noprint;
    define status        / display style={just=left
cellwidth=7CM}'';
    define _tot1          / display style={just=d
cellwidth=0.6cm} style(header)={just=center} "";
    define _tot2          / display style={just=d
cellwidth=0.6cm} style(header)={just=center} "";
    define _tot3          / display style={just=d
cellwidth=0.6cm} style(header)={just=center} "";
    define _tot4          / display style={just=d
cellwidth=0.6cm} style(header)={just=center} "";

```

```

        define totoverall/ display style={just=d cellwidth=0.6cm}
style(header)={just=center} "";
        define p1          / display style={just=r cellwidth=1.42cm}
style(header)={just=c}"";
        define p2          / display style={just=r cellwidth=1.42cm}
style(header)={just=c}"";
        define p3          / display style={just=r cellwidth=1.42cm}
style(header)={just=c}"";
        define p4          / display style={just=r cellwidth=1.42cm}
style(header)={just=c}"";
        define p_ov        / display style={just=r cellwidth=1.42cm}
style(header)={just=c}"";

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

break after page / page;

compute status;
    if statusn=1 or statusn=10 then
        call define(_row_, 'style', 'style=[font_weight=bold]');
endcomp;

    compute after page/style={just=left cellwidth=5cm
protectspecialchars=off};
        line "&linebot" ;
endcomp;

    compute before page / style={just=left 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};
        line 'Note: mCC = menthol conventional cigarettes; NRT gum =
Nicotine Replacement Therapy gum; THS = Tobacco Heating System.';
        line 'Note: Enrolled subjects are those who are eligible on
Day -1; Discontinued refers to randomized subjects who discontinued the
study before the planned';
        line 'discharge at Day 4; Completed refers to randomized
subjects who did not discontinue the study before the planned discharge
at Day 4.';
        line 'Note: Percentages are based on the number of subjects
indicated in the column header (N), apart from the Overall column where
percentages of completed and ';
        line 'discontinued refer to the total of subjects
randomized.';
        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_01.lst" new;
run;

proc contents data = table.T_15_02_01_01 varnum;
run;
ods listing close;

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

```