

/*****

** STUDY ID : 000000106343

**

** PROGRAM NAME : t_spiro.sas

**

** DATE : 19May2015

**

** PROGRAMMER : cvn_aramasah

**

** PURPOSE : QC the table Summary of full lung function results - safety
population (t_15_2_6_22)

**

** INPUT DATA : ADAM.ADSL, ADAM.ADXP

**

** OUTPUT DATA :

**

** SAS MACROS USED :

**

** MODIFICATIONS : DATE : MODIFIED BY : NOTES :

**

**

** PROGRAMMED USING SAS VERSION 9.3 **

** COPYRIGHT (C) 2015 BY COVANCE, PRINCETON NJ - USA - ALL RIGHTS RESERVED **

-----/

options notes source source2 nofullstimer validvarname=upcase missing=' ';

```

ods _all_ close;

ods listing;

%m_printto;

options notes nosource replace;

proc datasets lib=work nolist memtype=data kill; quit;


*=====;

* START OF PROGRAM CODE                                ;

*=====;


%let tflno=T_15_02_06_22;

%let TFLprg=t_spiro.sas;

%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 ;

*****
,


/*Use ADSL to get N numbers for column headers*/

```

```

data adsl;

    set adam.adsl;

    if trt01a='THSm2.2' then trt01an=1;

    if trt01a='mCC' then trt01an=2;

    if trt01a='SA' then trt01an=3;

        where safaf1 = 'Y';

        if missing(trt01an) then delete;

        if index(trt01a,'Exposed') then delete;

    output;

    trt01an=99;

    trt01a='Overall Safety';

    output;

run;

```

```

proc freq data=adsl noprint;

    table trt01an*trt01a/ out =tot(drop=percent);

run;

```

```

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

    attrib trt01a length =$40.

                trt01an length=8.;

    trt01an=1;

    trt01a='THSm2.2';

```

```
output;  
trt01an=2;  
trt01a='mCC';  
output;  
trt01an=3;  
trt01a='SA';  
output;  
trt01an=99;  
trt01a='Overall Safety';  
output;  
run;
```

```
data dumtrts2; /*Use this to output any columns for which N=0*/
```

```
attrib trta length =$40.
```

```
trtan length=8.;
```

```
trtan=1;  
trta='THSm2.2';  
output;  
trtan=2;  
trta='mCC';  
output;  
trtan=3;  
trta='SA';  
output;  
trtan=99;
```

```

        trta='Overall Safety';

        output;

run;

data tot2a;

        merge tot(in=a) dumtrts(in=b);

        by trt01an trt01a;

        if a or b;

        if b and not a then count=0;

        call symput('trt' || compress(put(trt01an,best.)), compress(put(count,best.)));

run;

```

```

data tot2;

        set tot2a;

        trta=trt01a;

        trtan=trt01an;

        drop trt01an trt01a;

run;

```

```

/*Bring in appropriate data from ADXP*/

```

```

data adxp1;

```

```

length avalc_ $40.;

```

```

/*      set adam.adxp(where=((paramcd in ('DFEVFVC' 'INTP' 'FEVMEAS' 'FVCMEAS' 'FEVPCT' 'FVCPCT'*/

```

```

/*      'WFEVMEAS' 'WFVCMEAS' 'WFVCPCT' 'WFEVPCT' 'WFEVFVC' 'WINTP') or (paramcd='FEVFVC' */

```

```
/*                                and xpstat ne 'NOT DONE')) and safafi = 'Y' and anl01fl='Y' and dtype ne  
'LOCF')); */
```

```
set adam.adxp;
```

```
*where=(paramn in (19, 10, 51, 54, 61, 64, 21, 11, 18, 9, 17, 8, 14, 5, 65, 13, 4, 6, 15, 20, 63)) and safafi =  
'Y'
```

```
and anl01fl='Y' and dtype ne 'LOCF';
```

```
where ((paramn in (19, 51, 54, 61, 64, 65, 63, 20) and anl01fl='Y') or paramn in (10, 11, 9, 8, 5, 4, 6))
```

```
and safafi = 'Y' and dtype ne 'LOCF';
```

```
/* */
```

```
    if upcase(avalc)='NORMAL' then avalc_='Normal';
```

```
        else if index(upcase(avalc),'ABNORMAL') gt 0 then avalc_='Abnormal';
```

```
/* */
```

```
    if trta='THSm2.2' then trtan=1;
```

```
    if trta='mCC' then trtan=2;
```

```
    if trta='SA' then trtan=3;
```

```
    if ablfl='Y' then do; avisit='Baseline'; avisitn=100; end;
```

```
    if avisit ne 'Baseline' and avisitn lt 101 then delete;
```

```
    if missing(trtan) then delete;
```

```
    if index(trta,'Exposed') then delete;
```

```
    output;
```

```
    trtan=99;
```

```
    trta='Overall Safety';
```

```

        output;

run;

/*INTP only*/

proc sort data=adxp1 out=adxp;

        by trtan trta avisitn avisit usubjid avalc_ paramn param paramcd xpclsig;

run;

proc freq data=adxp(where=((paramcd='INTP' and anl01fl='Y') or paramcd='WINTP')) noprint;

        table trtan*trta*avisitn*avisit*aval*avalc_*paramn*param*paramcd*xpclsig / out
=intp1(drop=percent);

run;

proc freq data=adxp(where=((paramcd='INTP' and anl01fl='Y') or paramcd='WINTP')) noprint;

        table trtan*trta*avisitn*avisit*paramn*param*paramcd / out =intp1_n(drop=percent); /* 7)
JMH 07Oct2014 */

run;

data intp1;

set intp1;

order=avisitn;

run;

data intp1_n;

set intp1_n;

```

```
order=avisitn-0.01;
```

```
avalc_='n';
```

```
run;
```

```
data intp1;
```

```
set intp1 intp1_n;
```

```
run;
```

```
proc sort data=intp1;by trtan trta paramn avisitn order;run;
```

```
data intp2 ;
```

```
merge intp1(in=a rename=(avalc_=avalc)) dumtrts2(in=b) tot2(rename=(count=total));
```

```
by trtan trta;
```

```
if a or b;
```

```
if b and not a then do;
```

```
count = 0;
```

```
avisitn=1;
```

```
avisit='Screening';
```

```
avalc='Normal';
```

```
end;
```

```
run;
```

```
proc sort data=intp2 nodupkey out=trtvis(keep=trtan trta avisitn avisit avalc paramn param paramcd);
```

```
by trtan trta avisitn avisit param paramcd;
```



```
run;
```

```
data dumrows;
```

```
set trtvis;
```

```
    attrib xpclsig length=$3.;
```

```
    avalc='Normal';
```

```
    xpclsig = '';
```

```
    output;
```

```
    avalc='Abnormal';
```

```
    xpclsig ='NCS';
```

```
    output;
```

```
    avalc='Abnormal';
```

```
    xpclsig = 'CS';
```

```
    output;
```

```
run;
```

```
proc sort data=dumrows;
```

```
    by trtan trta avisitn avisit param paramn paramcd avalc xpclsig;
```

```
run;
```

```
proc sort data=intp2;
```

```
    by trtan trta avisitn avisit param paramn paramcd avalc xpclsig;
```

```
run;
```

```

data intp3;

merge intp2(in=a) dumrows(in=b);

by trtan trta avisitn avisit param paramn paramcd avalc xpclsig;

if a or b;

if a then data=1;

if b then dummy=1;


attrib statval length=$20.

/*count1 length=$3.*;/

if b and not a then do;

    count=0;

end;


attrib paramc length = $100.

/*visit length = $100.* /

stat length = $100.;

paramc=strip(param);


if avisitn=1 then do; ord=1; *visit=avisit; end;

else if avisitn=100 then do; ord=5; *visit=avisit; end;

else if avisitn=106 then do; ord=7; *visit=avisit; end;

else if avisitn=191 then do; ord=8; *visit=avisit; end;

/*
else put "WA" "RNING: Unexpected value for avisitn: " avisitn=;*/

if upcase(avalc)='N' then do;

    stat='n';

```

```

        statord=1;
    end;
else if upcase(avalc)='NORMAL' then do;
        stat='Normal';
        statord=2;
    end;
else if index(upcase(avalc),'ABNORMAL') gt 0 and xpclsig='NCS' then do;
        stat='Abnormal NCS';
        statord=3;
    end;
else if index(upcase(avalc),'ABNORMAL') gt 0 and xpclsig='CS' then do;
        stat='Abnormal CS';
        statord=4;
    end;
/*      else put "WAR" "NING unexpected avalc/xpclsig" avalc= xpclsig=; */
statval=strip(put(count,best.)) ;

run;

proc sort data=intp3;
    by paramn paramc paramcd ord avisitn avisit statord stat avalc;
run;

proc transpose data=intp3 out=intp4(where=(stat ne 'DUMMY')) prefix=t;
    by paramn paramc paramcd ord avisitn avisit statord stat avalc;

```

```
var statval;  
id trtan;  
idlabel trta;
```

```
run;
```

```
data intp_perc;
```

```
set intp4;
```

```
where stat='n';
```

```
t1_=input(t1,best.);
```

```
t2_=input(t2,best.);
```

```
t3_=input(t3,best.);
```

```
t99_=input(t99,best.);
```

```
keep paramn avisitn t1_ t2_ t3_ t99_;
```

```
run;
```

```
proc sort data=intp4;by paramn avisitn;proc sort data=intp_perc;by paramn avisitn;run;
```

```
data intp5;
```

```
merge intp4 intp_perc;
```

```
by paramn avisitn;
```

```
if stat ne ";
```

```
if t1="" then t1='0';
```

```
if t2="" then t2='0';
```

```
if t3="" then t3='0';
```

```
if t99="" then t99='0';
```

```
run;
```

```
/*End of INTP, will set on with rest of data later*/
```

```
/*Everything except INTP*/
```

```
data adxp_orig;
```

```
set adxp(where=(paramcd not in ('INTP' 'WINTP' 'FEVPRED' 'FVCPRED' 'WFEVPRED'  
'WFVCPRED')));
```

```
if avisitn=1 then ord=1;
```

```
else if avisitn=100 then ord=5;
```

```
else if avisitn=106 then ord=7;
```

```
else if avisitn=191 then ord=12;
```

```
/* else put "WA" "RNING: Unexpected value for avisitn: " avisitn=;*/
```

```
statval=aval;
```

```
run;
```

```
data adxp_chg;
```

```
set adxp(where=(avisitn in(106 191)and paramcd not in ('INTP' 'WINTP' 'FEVPRED' 'FVCPRED'  
'WFEVPRED' 'WFVCPRED'))); /*Only keep days after baseline*/
```

```
if avisitn=106 then ord=7;/*Change from Baseline to Discharge confinement*/
```

```
else if avisitn=191 then ord=12;/*Change from Baseline to Discharge ambulatory*/
```

```
/* else put "WA" "RNING: Unexpected value for avisitn: " avisitn=;*/
```

```
statval=chg;
```

```
run;
```

```
/*Transpose for raw values*/
```

```
proc sort data=adxp_orig;
```

```
    by trtan trta paramn paramcd ord param avalu avisitn avisit;
```

```
run;
```

```
proc means data=adxp_orig noprint;
```

```
    var statval;
```

```
    by trtan trta paramn paramcd ord param avalu avisitn avisit;
```

```
    output out=results01_orig (drop=_TYPE_ _FREQ_) n=n1o mean=mean1o std=std1o  
    median=med1o min=min1o max=max1o lclm=lcio uclm=ucio;
```

```
run;
```

```
/*proc univariate data=adxp_orig noprint;*/
```

```
/*    var statval;*/
```

```
/*    by trtan trta paramn paramcd ord param avalu avisitn avisit; */
```

```
/*    output out=results01_orig n=n1o mean=mean1o std=std1o median=med1o min=min1o  
    max=max1o;*/
```

```
/*run;*/
```

```
/*Transpose for change from baseline values*/
```

```
proc sort data=adxp_chg;
```

```
    by trtan trta paramn paramcd ord param avalu avisitn avisit;
```

```
run;
```

```
proc means data=adxp_chg noprint;
```

```
    var statval;
```

```

        by trtan trta paramn paramcd ord param avalu avisitn avisit;

        output out=results01_chg(drop=_TYPE__FREQ_) n=n1c mean=mean1c std=std1c
median=med1c min=min1c max=max1c lclm=lcic uclm=ucic;

run;

/*proc univariate data=adxp_chg noprint;*/

/*    var statval;*/

/*    by trtan trta paramn paramcd ord param avalu avisitn avisit; */

/*    output out=results01_chg n=n1c mean=mean1c std=std1c median=med1c min=min1c
max=max1c;*/

/*run;*/

data results01;

    merge results01_orig results01_chg;

    by trtan trta paramn paramcd ord param avalu avisitn avisit;

run;

data results02;

    set results01;

        attrib meano length=$10.

                mino length=$15.

                no    length=$10.

                mediano length=$10.

                stdo length=$10.

                l_ci_u_o length=$20.;

```

if paramn in (14 17 5 8) then do;

no = left(compress(put(n1o,8.)));

if not missing(med1o) then mediano =
left(compress(put(ROUND(med1o,0.1),8.1))); /* 1) JH 22OCT2014 */

if not missing(mean1o) then meano =
left(compress(put(ROUND(mean1o,0.1),8.1))); /* 1) JH 22OCT2014 */

if not missing(std1o) then stdo=' ' || compress(put(0.01*ceil(std1o/0.01),8.2))
|| '');

if not missing(min1o) and not missing(max1o) then mino =
left(compress(put(ROUND(min1o,1),8.))) || ', ' || left(compress(put(ROUND(max1o,1),8.))); /* 1) JH
22OCT2014 */

if not missing(lcio) and not missing(ucio) then
l_ci_u_o=compress(put(0.1*floor(lcio/0.1),8.1)) || ', ' || compress(put(0.1*ceil(ucio/0.1),8.1));

attrib meanc length=\$10.

minc length=\$15.

nc length=\$10.

medianc length=\$10.

stdc length=\$10.

l_ci_u_c length=\$20.;

nc = left(compress(put(n1c,8.)));

if not missing(med1c) then medianc =
left(compress(put(ROUND(med1c,0.1),8.1))); /* 1) JH 22OCT2014 */

if not missing(mean1c) then meanc =
left(compress(put(ROUND(mean1c,0.1),8.1))); /* 1) JH 22OCT2014 */

if not missing(std1c) then stdc=' ' || compress(put(0.01*ceil(std1c/0.01),8.2)) ||
');


```
        if not missing(min1c) and not missing(max1c) then minc =  
left(compress(put(ROUND(min1c,1),8.))) || ', ' || left(compress(put(ROUND(max1c,1),8.))); /* 1) JH  
22OCT2014 */
```

```
        if not missing(lcic) and not missing(ucic) then  
l_ci_u_c=compress(put(0.1*floor(lcic/0.1),8.1)) || ', ' || compress(put(0.1*ceil(ucic/0.1),8.1));  
  
end;
```

```
else if paramn not in (14 17 5 8) then do;
```

```
        no = left(compress(put(n1o,8.)));
```

```
        if not missing(med1o) then mediano =  
left(compress(put(ROUND(med1o,0.001),8.3))); /* 1) JH 22OCT2014 */
```

```
        if not missing(mean1o) then meano =  
left(compress(put(ROUND(mean1o,0.001),8.3))); /* 1) JH 22OCT2014 */
```

```
        if not missing(std1o) then stdo=' ' ||  
compress(put(0.0001*ceil(std1o/0.0001),8.4)) || ' ';
```

```
        if not missing(min1o) and not missing(max1o) then mino =  
left(compress(put(ROUND(min1o,0.01),8.2))) || ', ' || left(compress(put(ROUND(max1o,0.01),8.2))); /* 1)  
JH 22OCT2014 */
```

```
        if not missing(lcio) and not missing(ucio) then l_ci_u_o=  
compress(put(0.001*floor(lcio/0.001),8.3)) || ', ' || compress(put(0.001*ceil(ucio/0.001),8.3));
```

```
        nc = left(compress(put(n1c,8.)));
```

```
        if not missing(med1c) then medianc =  
left(compress(put(ROUND(med1c,0.001),8.3))); /* 1) JH 22OCT2014 */
```

```
        if not missing(mean1c) and not missing(std1c) then meanc =  
left(compress(put(ROUND(mean1c,0.001),8.3))); /* 1) JH 22OCT2014 */
```

```
        if not missing(std1c) then stdc=' ' ||  
compress(put(0.0001*ceil(std1c/0.0001),8.4)) || ' ';
```

```
        if not missing(min1c) and not missing(max1c) then minc =  
left(compress(put(ROUND(min1c,0.01),8.2))) || ', ' || left(compress(put(ROUND(max1c,0.01),8.2))); /*  
1) JH 22OCT2014 */
```

```

                                if not missing(lcic) and not missing(ucic) then l_ci_u_c=
compress(put(0.001*floor(lcic/0.001),8.3))||', '|| compress(put(0.001*ceil(ucic/0.001),8.3));

end;

                                drop n1o mean1o std1o med1o min1o max1o n1c mean1c std1c med1c min1c
max1c lcio ucio lcic ucic;

/*  if meansdo='-0.0' then meansdo=tranwrd(meansdo,'-0.0','0.0'); */

/*                                if meansdc='-0.0' then meansdc=tranwrd(meansdc,'-0.0','0.0'); */

run;

data results03; /*Create text as required in output*/

                                set results02;

                                attrib paramc length = $100.

                                visit length = $100.;

/*  if param='Ratio between FEV1/FVC (Derived)' then param='Calculated ratio between
FEV1/FVC`{super 1}'; */

                                if avalu ne " then paramc=strip(param)||'('||strip(avalu)||')';

                                else paramc=strip(param);

                                if ord=1 then visit=avisit;

                                else if ord=5 then visit=avisit;

                                else if ord=7 then visit=avisit;

                                else if ord=12 then visit=avisit;

/*                                else put "WA" "RNING: Unexpected value for ord: " ord=;*/

run;

```

```
proc sort data=results03;

    by paramn paramc paramcd ord visit avisitn;

run;
```

```
proc transpose data=results03 out=results04_orig1 prefix=o name=varname;

    by paramn paramc paramcd ord visit avisitn;

    var no meano stdo mediano mino L_CI_U_O;

    id trtan;

    idlabel trta;

run;
```

```
data results04_orig;

length _o1 _o2 _o3 _o99 $20.;

    set results04_orig1;

    _o1=o1;

    _o2=o2;

    _o3=o3;

    _o99=o99;

    varname=tranwrd(varname,'O','C');

    drop o1 o2 o3 o99;

run;
```

```
proc transpose data=results03 out=results04_chg prefix=c name=varname;

    by paramn paramc paramcd ord visit avisitn;

    var nc meanc stdc medianc minc L_CI_U_c;
```

```
id trtan;

idlabel trta;

run;


proc sort data=results04_orig;

    by paramn paramc ord avisitn visit varname;

run;


proc sort data=results04_chg;

    by paramn paramc paramcd ord avisitn visit varname;

run;


data results04_chg;

    length _c1 _c2 _c3 _c99 $20.;

    set results04_chg;

    _c1=c1;

    _c2=c2;

    _c3=c3;

    _c99=c99;

    drop c1 c2 c3 c99;

run;


data results04;

    merge results04_orig results04_chg;

    by paramn paramc paramcd ord avisitn visit varname;
```

```
run;
```

```
data results05;
```

```
    set results04;
```

```
    attrib stat length = $100.;
```

```
    if varname='NC' then do; statord=1; stat='n'; end;
```

```
    else if varname='MEANC' then do; statord=2; stat='Mean'; end;
```

```
    else if varname='STDC' then do; statord=3; stat='(SD)'; end;
```

```
    else if varname='MEDIANC' then do; statord=5; stat='Median'; end;
```

```
    else if varname='MINC' then do; statord=6; stat='Min, Max'; end;
```

```
    else if varname='L_CI_U_C' then do; statord=4; stat='95% CI of Mean'; end;
```

```
    drop varname;
```

```
run;
```

```
data results06;
```

```
    set results05;
```

```
    if stat='n' and statord=1 then do;
```

```
        if missing(_o1) then _o1='0';
```

```
        if missing(_o2) then _o2='0';
```

```
        if missing(_o3) then _o3='0';
```

```
        if missing(_o99) then _o99='0';
```

```
    end;
```

```
run;
```

```
/*Now combine the stats with the classification results*/
```

```
data allresults;
```

```
    set results06 intp5;
```

```
    if paramn=10 then paramn=0.5;
```

```
    else if paramn=19 then paramn=22;
```

```
/*for interpretation values*/
```

```
    if paramn in(0.5 22) then do;
```

```
        _o1=t1;
```

```
        _o2=t2;
```

```
        _o3=t3;
```

```
        _o99=t99;
```

```
    end;
```

```
/*    visit=tranwrd(visit,'/',' ');*/
```

```
    drop t1 t2 t3 t99;
```

```
run;
```

```
data labels;
```

```
    set allresults;
```

```
    attrib      _o1 _o2 _o3 _o99 label = "Raw value"
```

```
               _c1 _c2 _c3 _c99 label = "Change";
```

```
_o1 = strip(_o1);  
_o2 = strip(_o2);  
_o3 = strip(_o3);  
_o99 = strip(_o99);  
_c1 = strip(_c1);  
_c2 = strip(_c2);  
_c3 = strip(_c3);  
_c99 = strip(_c99);
```

```
if index(stat,'ormal') then fl=1;
```

```
else fl=2;
```

```
if paramn in(0.5 22) then with=2;
```

```
else if index(paramc,'with') and index(paramc,'Interp')=0 then with=0;
```

```
else with=1;
```

```
if paramc='Interpretation' then paramn=0;
```

```
if avisit ne '' then visit=avisit;
```

```
place=2;
```

```
drop avisit;
```

```
if paramcd in ('INTP' 'WINTP') and avalc ne 'n' then do;
```

```
if compress(_o1) ne '0' then _o1=strip(_o1)||' ('|| strip(put(((input(_o1,best.))/t1_*100),5.1))||'%');
```

```
if compress(_o2) ne '0' then _o2=strip(_o2)||' ('|| strip(put(((input(_o2,best.))/t2_*100),5.1))||'%');
```

```
if compress(_o3) ne '0' then _o3=strip(_o3)||' ('|| strip(put(((input(_o3,best.))/t3_*100),5.1))||'%');
```

```
if compress(_o99) ne '0' then _o99=strip(_o99)||' ('||  
strip(put(((input(_o99,best.))/t99_*100),5.1))||'%');
```

```

end;

run;

/* pre-randomization */

/*Use ADSL to get N numbers for column headers*/
data adsl_pre;

    set adam.adsl;

    if trt01a='THSm2.2' then trt01an=1;

    if trt01a='mCC' then trt01an=2;

    if trt01a='SA' then trt01an=3;

        where safbfl = 'Y' ;

        if missing(trt01an) then delete;

        if index(trt01a,'Exposed') then delete;

    output;

    trt01an=99;

    trt01a='Overall Safety';

    output;

run;

proc freq data=adsl_pre noprint;

    table trt01an*trt01a/ out =tot_pre(drop=percent);

run;

```



```
data dumtrts_pre; /*Use this to output any columns for which N=0*/
```

```
attrib trt01a length =$40.
```

```
trt01an length=8.;
```

```
trt01an=1;
```

```
trt01a='THSm2.2';
```

```
output;
```

```
trt01an=2;
```

```
trt01a='mCC';
```

```
output;
```

```
trt01an=3;
```

```
trt01a='SA';
```

```
output;
```

```
trt01an=96;
```

```
trt01a='Product Test';
```

```
output;
```

```
trt01an=99;
```

```
trt01a='Overall Safety';
```

```
output;
```

```
run;
```

```
data dumtrts2_pre; /*Use this to output any columns for which N=0*/
```

```
attrib trta length =$40.
```

```
trtan length=8.;
```

```
trtan=1;

trta='THSm2.2';

output;

trtan=2;

trta='mCC';

output;

trtan=3;

trta='SA';

output;

trtan=96;

trta='Product Test';

output;

trtan=99;

trta='Overall Safety';

output;

run;
```

```
data tot2a_pre;

    merge tot_pre(in=a) dumtrts_pre(in=b);

    by trt01an trt01a;

    if a or b;

    if b and not a then count=0;

    call symput('pre_trt' || compress(put(trt01an,best.)), compress(put(count,best.)));

run;
```

```
%put &pre_trt99.;
```

```
data tot2_pre;
```

```
    set tot2a_pre;
```

```
    trta=trt01a;
```

```
    trtan=trt01an;
```

```
    drop trt01an trt01a;
```

```
run;
```

```
/*Bring in appropriate data from ADXP*/
```

```
data adxp1_pre;
```

```
length avalc_ $40.;
```

```
    *set adam.adxp(where=/*(paramcd in ('DFEVFVC' 'INTP' 'FEVMEAS' 'FVCMEAS' 'FEVPCT'  
'FVCPCT'
```

```
    'WFEVMEAS' 'WFVCMEAS' 'WFVCPCT' 'WFEVPCT' 'WFEVFVC' 'WINTP') or (paramcd='FEVFVC'  
    and xpstat ne 'NOT DONE')) and */*((safbfl = 'Y' and anl01fl='Y') or  
substr(paramcd,1,1)='W') and avisitn in (1,100) and dtype ne 'LOCF'));
```

```
    *set adam.adxp(where=((paramn in (19, 10, 51, 54, 61, 64, 21, 11, 18, 9, 17, 8, 14, 5, 65, 13, 4, 6,  
15))
```

```
    and safbfl = 'Y' and anl01fl='Y' and avisitn in (1,  
100) and dtype ne 'LOCF'));
```

```
set adam.adxp (where=((((paramn in (19, 51, 54, 61, 64, 21, 18, 17, 14, 65, 13, 15 20 63) and anl01fl='Y')  
    or (paramn in (10, 11, 9, 8, 5, 4, 6)))
```

```
    and safbfl = 'Y' and avisitn in (1, 100) and dtype ne 'LOCF'))); /* don't consider ANL01FL  
for PARAMCDs that start with W */
```

```

/* */

if upcase(avalc)='NORMAL' then avalc_='Normal';

    else if index(upcase(avalc),'ABNORMAL') gt 0 then avalc_='Abnormal';

/* */

if trta='THSm2.2' then trtan=1;

if trta='mCC' then trtan=2;

if trta='SA' then trtan=3;

/* if abfl='Y' then do; avisit='Baseline'; avisitn=100; end; */

/* if avisit ne 'Baseline' and avisitn lt 101 then delete; */

if missing(trtan) then delete;

if index(trta,'Exposed') then delete;

output;

trtan=99;

trta='Overall Safety';

output;

run;

/*INTP only*/

proc sort data=adxp1_pre out=adxp_pre;

    by trtan trta avisitn avisit usubjid avalc_ paramn param paramcd xpclsig;

```

```
run;
```

```
proc freq data=adxp_pre(where=( (paramcd in ('INTP') and anl01fl='Y') or paramcd='WINTP')) noprint;
```

```
    table trtan*trta*avisitn*avisit*aval*avalc_*paramn*param*paramcd*xpclsig / out  
=intp1_pre(drop=percent);
```

```
run;
```

```
proc freq data=adxp_pre(where=( (paramcd in ('INTP') and anl01fl='Y') or paramcd='WINTP')) noprint;
```

```
    table trtan*trta*avisitn*avisit*paramn*param*paramcd / out =intp1_pre_n(drop=percent); /*  
7) JMH 07Oct2014 */
```

```
run;
```

```
data intp1_pre;
```

```
set intp1_pre;
```

```
order=avisitn;
```

```
run;
```

```
data intp1_pre_n;
```

```
set intp1_pre_n;
```

```
order=avisitn-0.01;
```

```
avalc_='n';
```

```
run;
```

```
data intp1_pre;
```

```
set intp1_pre intp1_pre_n;
```

```
run;
```

```
proc sort data=intp1_pre;by trtan trta paramn avisitn order;run;
```

```
data intp2_pre ;
```

```
merge intp1_pre(in=a rename=(avalc_=avalc)) dumtrts2_pre(in=b) tot2_pre  
(rename=(count=total));
```

```
by trtan trta;
```

```
if a or b;
```

```
if b and not a then do;
```

```
count = 0;
```

```
avisitn=1;
```

```
avisit='Screening';
```

```
avalc='Normal';
```

```
end;
```

```
run;
```

```
proc sort data=intp2_pre nodupkey out=trtvis_pre(keep=trtan trta avisitn avisit avalc paramn param  
paramcd);
```

```
by trtan trta avisitn avisit param paramcd;
```

```
run;
```

```
data dumrows_pre;
```

```
set trtvis_pre;
```

```
attrib xpclsig length=$3.;
```

```
avalc='Normal';
```

```

        xpclsig = "";

        output;

        avalc='Abnormal';

        xpclsig ='NCS';

        output;

        avalc='Abnormal';

        xpclsig = 'CS';

        output;

run;


proc sort data=dumrows_pre;

        by trtan trta avisitn avisit param paramn paramcd avalc xpclsig;

run;


proc sort data=intp2_pre;

        by trtan trta avisitn avisit param paramn paramcd avalc xpclsig;

run;


data intp3_pre;

        merge intp2_pre(in=a) dumrows_pre(in=b);

        by trtan trta avisitn avisit param paramn paramcd avalc xpclsig;

        if a or b;

        if a then data=1;

        if b then dummy=1;

```

```
attrib statval length=$20.
```

```
/*count1 length=$3.*/;
```

```
if b and not a then do;
```

```
count=0;
```

```
end;
```

```
attrib paramc length = $100.
```

```
/*visit length = $100.*/
```

```
stat length = $100.;
```

```
paramc=strip(param);
```

```
if avisitn=1 then do; ord=1; *visit=avisit; end;
```

```
else if avisitn=100 then do; ord=5; *visit=avisit; end;
```

```
/* else put "WA" "RNING: Unexpected value for avisitn: " avisitn=;*/
```

```
if upcase(avalc)='N' then do;
```

```
stat='n';
```

```
statord=1;
```

```
end;
```

```
else if upcase(avalc)='NORMAL' then do;
```

```
stat='Normal';
```

```
statord=2;
```

```
end;
```

```
else if index(upcase(avalc),'ABNORMAL') gt 0 and xpclsig='NCS' then do;
```

```
stat='Abnormal NCS';
```

```
statord=3;
```



```

        end;

        else if index(upcase(avalc),'ABNORMAL') gt 0 and xpclsig='CS' then do;

                stat='Abnormal CS';

                statord=4;

        end;

/*        else put "WAR" "NING unexpected avalc/xpclsig" avalc= xpclsig=; */

        statval=strip(put(count,best.)) ;

run;


proc sort data=intp3_pre;

        by paramn paramc paramcd ord avisitn avisit statord stat avalc;

run;


proc transpose data=intp3_pre out=intp4_pre (where=(stat ne 'DUMMY')) prefix=t;

        by paramn paramc paramcd ord avisitn avisit statord stat avalc;

        var statval;

        id trtan;

        idlabel trta;

run;


data intp_pre_perc;

set intp4_pre;

where stat='n';

t1_=input(t1,best.);

```

```

t2_=input(t2,best.);
t3_=input(t3,best.);
t96_=input(t96,best.);
t99_=input(t99,best.);
keep paramn avisitn t1_ t2_ t3_ t99_ t96_;

run;

proc sort data=intp4_pre;by paramn avisitn;proc sort data=intp_pre_perc;by paramn avisitn;run;

data intp5_pre;
merge intp4_pre intp_pre_perc;
by paramn avisitn;

/**/

/*data intp5_pre;*/

/* set intp4_pre;*/

        if stat ne "";

        if t1="" then t1='0';

        if t2="" then t2='0';

        if t3="" then t3='0';

        if t96="" then t96='0';

        if t99="" then t99='0';

run;

/*End of INTP, will set on with rest of data later*/

```

```

/*Everything except INTP*/

data adxp_orig_pre;

    set adxp_pre(where=(paramcd not in ('INTP' 'WINTP' /*'FEVPRED' 'FVCPRED' 'WFEVPRED'
'WFVCPRED'*/)));

    if avisitn=1 then ord=1;

    else if avisitn=100 then ord=5;

/*    else put "WA" "RNING: Unexpected value for avisitn: " avisitn=;*/

    statval=aval;

run;


/*Transpose for raw values*/

proc sort data=adxp_orig_pre;

    by trtan trta paramn paramcd ord param avalu avisitn avisit;

run;


proc means data=adxp_orig_pre noprint;

    var statval;

    by trtan trta paramn paramcd ord param avalu avisitn avisit;

    output out=results01_orig_pre n=n1o mean=mean1o std=std1o median=med1o min=min1o
max=max1o lclm=lcio uclm=ucio;

run;


/*proc univariate data=adxp_orig_pre noprint;*/

/*    var statval;*/

/*    by trtan trta paramn paramcd ord param avalu avisitn avisit; */

```

```
/*      output out=results01_orig_pre n=n1o mean=mean1o std=std1o median=med1o min=min1o
max=max1o;*/
```

```
/*run;*/
```

```
data results01_pre;
```

```
    set results01_orig_pre;
```

```
        attrib meano length=$10.
```

```
                mino length=$15.
```

```
                no    length=$10.
```

```
                mediano length=$10.
```

```
                stdo length=$10.
```

```
                l_ci_u_o length=$20.;
```

```
if paramn in (14 17 5 8 ) then do;
```

```
        no = left(compress(put(n1o,8.)));
```

```
        if not missing(med1o) then mediano =
left(compress(put(ROUND(med1o,0.1),8.1))); /* 1) JH 22OCT2014 */
```

```
        if not missing(mean1o) then meano =
left(compress(put(ROUND(mean1o,0.1),8.1))); /* 1) JH 22OCT2014 */
```

```
        if not missing(std1o) then stdo='(' || compress(put(0.01*ceil(std1o/0.01),8.2))
|| ')';
```

```
        if not missing(min1o) and not missing(max1o) then mino =
left(compress(put(ROUND(min1o,1),8.))) || ', ' || left(compress(put(ROUND(max1o,1),8.))); /* 1) JH
22OCT2014 */
```

```
        if not missing(lcio) and not missing(ucio) then
l_ci_u_o=compress(put(0.1*floor(lcio/0.1),8.1)) || ', ' || compress(put(0.1*ceil(ucio/0.1),8.1));
```

```
end;
```

```

else do;

    no = left(compress(put(n1o,8.)));

    if not missing(med1o) then mediano =
left(compress(put(ROUND(med1o,0.001),8.3))); /* 1) JH 22OCT2014 */

    if not missing(mean1o) then meano =
left(compress(put(ROUND(mean1o,0.001),8.3))); /* 1) JH 22OCT2014 */

    if not missing(std1o) then stdo=' ' ||
compress(put(0.0001*ceil(std1o/0.0001),8.4)) || ' ';

    if not missing(min1o) and not missing(max1o) then mino =
left(compress(put(ROUND(min1o,0.01),8.2))) || ', ' || left(compress(put(ROUND(max1o,0.01),8.2)));/* 1)
JH 22OCT2014 */

    if not missing(lcio) and not missing(ucio) then
l_ci_u_o=compress(put(0.001*floor(lcio/0.001),8.3))||', '||compress(put(0.001*ceil(ucio/0.001),8.3));

end;

    drop n1o mean1o std1o med1o min1o max1o lcio ucio;

/* if meansdo='-0.0' then meansdo=tranwrd(meansdo,'-0.0','0.0'); */

/* if meansdc='-0.0' then meansdc=tranwrd(meansdc,'-0.0','0.0'); */

run;

data results02_pre; /*Create text as required in output*/

    set results01_pre;

    attrib paramc length = $100.

        visit length = $100.;

/* if param='Ratio between FEV1/FVC (Derived)' then param='Calculated ratio between
FEV1/FVC`{super 1}`'; */

```

```

if avalu ne " then paramc=strip(param)||'('||strip(avalu)||');
    else paramc=param;

if ord=1 then visit=avisit;

else if ord=5 then visit=avisit;

/*    else put "WA" "RNING: Unexpected value for ord: " ord=;*/

run;

proc sort data=results02_pre;

    by paramn paramc paramcd ord visit avisitn;

run;

proc transpose data=results02_pre out=results03_pre prefix=o name=varname;

    by paramn paramc paramcd ord visit avisitn;

var no meano stdo mediano mino l_ci_u_o;

id trtan;

idlabel trta;

run;

data results03_pre;

    length _o1 _o2 _o3 _o96 _o99 $20.;

    set results03_pre;

    _o1=o1;

    _o2=o2;

    _o3=o3;

```

```
_o96=o96;  
_o99=o99;  
varname=tranwrd(varname,'O','C');  
drop o1 o2 o3 o96 o99;  
run;
```

```
proc sort data=results03_pre;  
by paramn paramc paramcd ord avisitn visit varname;  
run;
```

```
data results04_pre;  
set results03_pre;  
attrib stat length = $100.;  
if varname='NC' then do; statord=1; stat='n'; end;  
else if varname='MEANC' then do; statord=2; stat='Mean'; end;  
else if varname='STDC' then do; statord=3; stat='(SD)'; end;  
else if varname='MEDIANC' then do; statord=5; stat='Median'; end;  
else if varname='MINC' then do; statord=6; stat='Min, Max'; end;  
else if varname='L_CI_U_C' then do; statord=4; stat='95% CI of Mean'; end;  
drop varname;  
run;
```

```
data results05_pre;
```

```

set results04_pre;

if stat='n' and statord=1 then do;

    if missing(_o1) then _o1='0';

    if missing(_o2) then _o2='0';

    if missing(_o3) then _o3='0';

    if missing(_O96) then _O96='0';

    if missing(_o99) then _o99='0';

end;

run;

/*Now combine the stats with the classification results*/

data allresults_pre;

set results05_pre intp5_pre;

if paramn=10 then paramn=0.5;

else if paramn=19 then paramn=22;

if avisit ne " then visit=avisit;

/*for interpretation values*/

if paramn in(0.5 22) then do;

    _o1=t1;

    _o2=t2;

    _o3=t3;

    _o96=t96;

```



```

        _o99=t99;

    end;

_o97=_o96;

    drop t1 t2 t3 t96 _o96 t99 avisit avalc ;

run;


data labels_pre;

    set allresults_pre;

    attrib          _o1 _o2 _o3 _o96 _o99 label = "Raw value";


                                _o1 = strip(_o1);

                                _o2 = strip(_o2);

                                _o3 = strip(_o3);

                                _o96 = _o97;

                                _o99 = strip(_o99);


if index(stat,'ormal') then fl=1;

else fl=2;


if paramn in(0.5 22) then with=2;

else if index(paramc,'with') and index(paramc,'Interp')=0 then with=0;

else with=1;

if paramc='Interpretation' then paramn=0;

/*if avisit ne " then visit=avisit;*/

place=1;

```

```

drop _o97;

if paramcd in ('WINTP' 'INTP') and stat ne 'n' then do;

if compress(_o1) ne '0' then _o1=strip(_o1)||' ('|| strip(put(((input(_o1,best.))/t1_*100),5.1))||'%');

if compress(_o2) ne '0' then _o2=strip(_o2)||' ('|| strip(put(((input(_o2,best.))/t2_*100),5.1))||'%');

if compress(_o3) ne '0' then _o3=strip(_o3)||' ('|| strip(put(((input(_o3,best.))/t3_*100),5.1))||'%');

if compress(_o96) ne '0' then _o96=strip(_o96)||' ('||
strip(put(((input(_o96,best.))/t96_*100),5.1))||'%');

if compress(_o99) ne '0' then _o99=strip(_o99)||' ('||
strip(put(((input(_o99,best.))/t99_*100),5.1))||'%');

end;

run;

/* end of pre-randomization */

```

```

proc sort data=labels;

by place paramn paramcd ord statord;

proc sort data=labels_pre;

by place paramn paramcd ord statord;

run;

```

```

data final;

set labels_pre labels;

if indexw(upcase(visit),'UNSCHEDULED') ne 0 then delete;

if visit ne "";

run;

data final;

```

```
set final;
```

```
    if
```

```
/* page numbers */
```

```
proc sort data=final out=page (keep=place paramn avisitn) nodupkey;
```

```
by place paramn avisitn;
```

```
run;
```

```
data page;
```

```
set page;
```

```
page=_n_;
```

```
run;
```

```
proc sort data=final;
```

```
by place paramn avisitn;
```

```
proc sort data=page;
```

```
by place paramn avisitn;
```

```
run;
```

```
data paging;
```

```
merge final page;
```

```
by place paramn avisitn;
```

```
    call symput("page",compress(put(page,best.)));
```

```
        if place=1 then category="Pre-Randomization Period";
```

```
        else if place=2 then category="Randomized Period";
```

```
flag=1;
```

```
/*      flag=ceil(_n_/8);*/
```

```
run;
```

```
proc sort data=paging;
```

```
by place paramn avisitn statord;
```

```
run;
```

```
data paging;
```

```
set paging;
```

```
by place paramn avisitn statord;
```

```
/*if first.avisitn then do;*/
```

```
/*visit=visit;*/
```

```
/*end;*/
```

```
/*else do;*/
```

```
/*visit=";*/
```

```
/*end;*/
```

```
run;
```

```
proc sort data=paging;
```

```
by paramn;
```

```
run;
```

```
data lastpage_1;
```

```
set paging;
```

```
by paramn;
```

```
if last.paramn;
```

```
run;
```

```
proc sort data=lastpage_1;
```

```
by paramn ord;
```

```
run;
```

```
data lastpage;
```

```
set lastpage_1;
```

```
by paramn ord;
```

```
if last.ord then call symput('maxpage', trim(left(put(page,best.))));
```

```
/*if last.ord then call symput('maxpage', trim(left(put(flag,best.))));*/
```

```
run;
```

```
proc sort data=paging;
```

```
by page paramn avisitn ord statord;
```

```
run;
```

```
proc sql noprint;
```

```
create table tflds.&tflno as
```

```
select category, paramcd,paramc, visit, stat, _o1 as THS_raw, _c1 as ths_chg, _o2 as mcc_raw, _c2 as  
mcc_chg, _o3 as sa_raw, _c3 as sa_chg, _o96 as product_test,_o99 as overall_raw, _c99 as overall_chg,  
paramn, page
```

```
from paging
```

```
order by place, paramn, ord, statord;
```

```
quit;
```

```
/* figure output */
```

```
data tflds.&tflno._F;
```

```
set results02_pre (keep=param paramn paramcd trtan trta avisitn avisit avalu meano mino L_CI_U_O  
in=_1)
```

```
results03 (keep=param paramn paramcd trtan trta avisitn avisit avalu meanc minc  
L_CI_U_C meano mino L_CI_U_O in=_2);
```

```
if _1 then category='Pre-Randomization';
```

```
if _2 then category='Randomization';
```

```
min_original=input(scan(mino,1,','),best.);
```

```
max_original=input(scan(mino,2,','),best.);
```

```
/*max_original=input(maxo,best.);*/
```

```
mean_original=input(meano,best.);
```

```
l_ci_original=input(scan(L_CI_U_O,1,','),best.);
```

```
u_ci_original=input(scan(L_CI_U_O,2,','),best.);
```

```
if minc ne " then min_change=input(scan(minc,1,','),best.);
```

```
if minc ne " then max_change=input(scan(minc,2,','),best.);
```

```
/*if maxc ne " then max_change=input(maxc,best.);*/
```

```
if meanc ne " then mean_change=input(meanc,best.);
```

```

if L_CI_U_C ne "" then l_ci_change=input(scan(L_CI_U_C,1,','),best.);
if L_CI_U_C ne "" then u_ci_change=input(scan(L_CI_U_C,2,','),best.);

drop mino /*maxo*/ meano minc /*maxc*/ meanc L_CI_U_O L_CI_U_C;

if indexw(upcase(avisit),'UNSCHEDULED') ne 0 then delete;

run;

/* end*/

data paging;

set paging;

by place paramn avisitn statord;

if first.avisitn then do;

visit=visit;

paramc=paramc;

end;

else do;

visit="";

paramc="";

end;

run;

proc sort data=paging;

by place paramn ord statord;

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.t106343 (read) ;

ods results off;

ods rtf toc_data file="/cvn/projects/prj/data/000000106343/TFL/dev/Tables/&tflno..rtf" style=t106343
startpage=yes headery=1440 footery=1440 ;

ods noproctitle;

%do i=1 %to &page;

title ;

footnote;

%let wd=0;

%let supfl=0;

%let npage=%eval(&i);
```



```

data comp;

    set paging /*(drop=page)*/ end=eof;

        where page=&i;

/*      where flag=&i;*/

        call symput("place",compress(put(place,best.)));

/*      call symput('paramn',STRIP(PUT(paramn,BEST.))); */


/* Amend title as needed */

        _firtitl="Table 15.2.6.22 Summary of Full Lung Function Results - Safety Population";

        _upcas=(length("Path: &TFLpath.")-
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;

        len=&blankn.-length("(Page &npage of &maxpage)");

        if eof then do;

            call symput('paramn',STRIP(PUT(paramn,BEST.)));

            if index(paramc,"super") then call symput('supfl',"1");

            call symput('_FSRTITL', trim(left(_firtitl)));

            call symput('perid', strip(category));

/*          %if &place.=1 %then %do;*/

/*          _SECTITL1 = "Safety Time Period: Pre-Randomized Period";*/

/*          %end;*/

/**/

/*          %else %if &place.=2 %then %do;*/

/*          _SECTITL2 = "Safety Time Period: Randomized Period";*/

/*          %end;*/

            call symput('_blankn', compress(put(len,best.)));

        end;

```

```
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 missing nowd split = '$';;
```

```
/*      column place flag page with paramn PARAMC avisitn ord visit statord stat */
```

```
      column place flag page with paramn ord statord PARAMC avisitn visit stat
```

```
      %if &place.=1 %then %do;
```

```
          ("THSm2.2 $(N=&pre_trt1.) &linebot" _O1 )
```

```
          ("mCC $(N=&pre_trt2.) &linebot" _o2 )
```

```
          ("SA $(N=&pre_trt3.) &linebot" _o3)
```

```
          ( "Product Use$(N=&pre_trt96.) &linebot" _o96)
```

```
          ( "Overall Safety$(N=&pre_trt99.) &linebot" _o99)
```

```
      %end;
```

```
      %if &place.=2 %then %do;
```

```
          %if &paramn.=0 or &paramn.=0.5 %then %do;
```

```
              ("THSm2.2 $(N=&trt1) &linebot" _O1 )
```

```
              ("mCC $(N=&trt2) &linebot" _o2 )
```

```

("SA $(N=&trt3) &linebot" _o3 )
( "Overall Safety$(N=&trt99) &linebot" _o99 )

%end;

%else %do;

("THSm2.2 $(N=&trt1) &linebot" _O1 _c1)

("mCC $(N=&trt2) &linebot" _o2 _c2)

("SA $(N=&trt3) &linebot" _o3 _c3)

( "Overall Safety$(N=&trt99) &linebot" _o99 _c99)

%end;

%end;;

;

define place          / order order=internal noprint;

define flag    / order order=internal noprint;

define page    / order order = internal noprint;

define with    / order order = internal noprint;

define paramn    / order order = internal noprint;

define avisitn    / order order=internal noprint;

define ord    / order order = internal noprint;

define statord    / order order = internal noprint;

define paramc    / group style={just=l cellwidth=0.5 cm} style(header)={just=left}
'Parameter$(units)';

define stat    / group display style={just=l cellwidth=0.3 cm}
style(header)={just=left}'Statistic'; /* 11) JMH 16Sep2014 */

%if &place.=1 %then %do;

```

```

define visit / group style={just=l cellwidth=0.3 cm} style(header)={just=left}
'Study$Day';

define _O1 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o2 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o3 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o96 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o99 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

%end;

%if &place.=2 %then %do;

```

```

%if &paramn.=0 or &paramn.=0.5%then %do;

define visit / group style={just=l cellwidth=0.3 cm} style(header)={just=left}
'Study$Day';

define _o1 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o2 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o3 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

define _o99 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

%end;

%else %do;

```

```

define visit / group style={just=l cellwidth=0.5 cm} style(header)={just=left}
'Study$Day';

define _o1 / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw$value';

```

```

define _c1      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left};

define _o2      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw$value';

define _c2      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left};

define _o3      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw$value';

define _c3      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left};

define _o99     / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw$value';

define _c99     / display style={just=left cellwidth=0.3 cm}
style(header)={just=left};

```

```

%end;

```

```

%end;

```

```

break before flag / page %if &i=1 %then %do;

```

```

contents="&_fsrtitl" %end; %else %do; contents=" %end;;

```

```

break after page / page;

```

```

/* break after flag / page;*/

```

```

compute after ord;

```

```

line " ";

```

```

endcomp;

```

```

/* compute after flag / style={protectspecialchars=off};;*/
/* line "&linetop";*/
/* endcomp;*/

compute before page / style={protectspecialchars=off};

line "&linetop";

endcomp;

compute before _page_ / style={just=left protectspecialchars=off};

%if &place.=1 %then %do;

line "\b\fs24\sa24&_FSRTITL." ;

line "&linebot";

line "Safety Time Period: Pre-Randomization Period";

%end;

%else %do;

line "\b\fs24\sa24&_FSRTITL." ;

line "&linebot";

line "Safety Time Period: Randomization Period";

%end;

endcomp;

compute after _page_ / style={just=left protectspecialchars=off pretext="&linetop."};

%if &place.=1 %then %do;

line "Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THS =
Tobacco Heating System.";

line "Note: NCS = not clinically significant; CS = clinically significant.";

line "Note: 'Product Test' refers to all subjects who tested the THS product on Day -2 but
were not randomized. The Overall Safety refers to all subjects exposed to THSm2.2.";

```

```
line 'Note: Change is change from baseline, where baseline is the last assessment prior  
to first product use in CC/THS 2.2 arms on Day 1 or last assessment prior to 08:00 AM in SA arm on Day  
1.';
```

```
line ' ';
```

```
line 'Appendix 15.3.6.11';
```

```
line "Study ID:ZRHM-REXA-08-US Program:&TFLprg Status: &status" &_blankn.*"\~\~"  
"&sysdate" &_blankn.*"\~\~" "(Page &i of &page)";
```

```
%end;
```

```
%else %do;
```

```
line "Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THS =  
Tobacco Heating System.";
```

```
line "Note: NCS = not clinically significant; CS = clinically significant."
```

```
line 'Note: Change is change from baseline, where baseline is the last assessment prior  
to first product use in CC/THS 2.2 arms on Day 1 or last assessment prior to 08:00 AM in SA arm on Day  
1.';
```

```
line ' ';
```

```
line 'Appendix 15.3.6.11';
```

```
line "Study ID:ZRHM-REXA-08-US Program:&TFLprg Status: &status" &_blankn.*"\~\~"  
"&sysdate" &_blankn.*"\~\~" "(Page &i of &page)";
```

```
%end;
```

```
endcomp;
```

```
run;
```

```
%end;
```

```
ods rtf close;
```

```
ods results on;
```

```
ods path sashelp.tmplmst (read);
```

```
%mend ;
```

```
%outrtf(blankn=40, halfblk=N);
```

```
ods listing;
```

```
ods listing close;
```

```
proc datasets lib=work nolist memtype=data kill; quit;
```

```
%m_logchk;
```

```
*=====;
```

```
* END OF PROGRAM CODE ;
```

```
*=====;
```