```
%macro mu_nobs(in_data, where=) / store SOURCE des='V1.0.0.1';

%*************************************************************************
FILENAME: MU_NOBS.SAS
DEVELOPER: (b) (6)
PLATFORM: PC SAS 9.1.3, 9.2
DATE: 20120207
PURPOSE:  This macro will return a macro variable in the form IN_DATA_NOBS
        containing the number of observations in IN_DATA
        Library references will be stripped from the IN_DATA_NOBS macro
        variable name.
        For example, %mu_nobs(trans.dmg) will return &dmg_nobs

        If a where clause is passed, the macro will return the number of
observations
        in the subsetted data set


MODIFIED BY:            DATE:
*************************************************************************;

%global mu_nobs_rc;
%let mu_nobs_rc = 0;
%if "&where" ne "" and %sysfunc(exist(&in_data)) %then %do;
        %if %sysfunc(exist(__mu_nobs__)) %then %do;
          %put ALERT_C: __MU_NOBS__ already exists.  No subsetting will be done. ;
          %put ALERT_C: Number of obs in &in_data (where &where) cannot be
determined  Macro will stop;
          %let mu_nobs_rc = 2;
          %goto exit;
        %end;
        data __mu_nobs__;
          set &in_data;
          where &where;
        run;
        %if %index(&in_data, .) gt 0 %then %let in_data = %scan(&in_data, 2, .);
        %global &in_data._nobs;
        %let dsid = %sysfunc(open(__mu_nobs__));
        %let &in_data._nobs = %sysfunc(attrn(&dsid, nobs));
        %let close = %sysfunc(close(&dsid));
        %put &SYSMACRONAME: %upcase(&in_data._nobs)=&&&in_data._nobs;

        proc datasets lib=work mt=data nolist;
          delete __mu_nobs__;
          run;
        quit;

%end;
%else %if %sysfunc(exist(&in_data)) %then %do;
        %let dsid = %sysfunc(open(&in_data));
```

```
        %if %index(&in_data, .) gt 0 %then %let in_data = %scan(&in_data, 2, .);
        %global &in_data._nobs;
        %let &in_data._nobs = %sysfunc(attrn(&dsid, nobs));
        %let close = %sysfunc(close(&dsid));
        %put &SYSMACRONAME: %upcase(&in_data._nobs)=&&&in_data._nobs;
%end;
%else %do;
        %put ALERT_I: %upcase(&in_data) does not exist;
        %let mu_nobs_rc = 1;
%end;

%exit:

%put MU_NOBS_RC = &mu_nobs_rc;
%mend mu_nobs;
```