

SAS doesn't provide many good solutions to produce CSV files from datasets. There are three main options: using a PROC EXPORT with DBMS=CSV, using the built-in macro %ds2csv, or using ODS CSV. None of these are very capable of producing properly formatted CSV files; PROC EXPORT does not apply quotation around character strings, %ds2csv applies quotes to *all* variables including numeric variables, and ODS CSV has a tendency to include many other things than just the data (which, yes, can be prevented).

The below is a short but working starting point for a macro that produces a CSV file that applies quotations around character variables and no quotation around numeric variables. It includes a header row with variables names and you can choose your delimiter.

```

/*****
Macro by OCS Life Sciences
*****/
Parameters :
    inds       : [M] dataset name including library (e.g. SASHELP.CLASS)
    outcsv     : [M] path and filename of CSV file
    delim      : [O] delimiter to use (default is a comma)

Sample call :
    %sasToCSV(sashelp.class, C:\Users\ABC\Desktop\class.csv, delim=%str(,));
*****/
%macro sasToCSV(inds, outcsv, delim=%str(,));

    %put =====;
    %put Start of macro sasToCSV;
    %put -----;
    %put The following macro parameters were used:;
    %put - INDS    : &inds.;
    %put - OUTCSV  : &outcsv;
    %put - DELIM   : &delim;
    %put =====;
    %put;

    /* Get the name, type, and varnum of all variables
       in the input dataset and sort them by order, which
       represents the order of appearance of the
       variables in the input dataset. */
    proc contents data=&inds.
        out =work._tempstc_contents (keep = type name varnum)
        noprint;

    run;

    proc sort data=work._tempstc_contents;
        by varnum;
    run;

    /* This step generates two macro variables: one that
       contains all variable names (comma separated) to be
       used as the header row in the CSV, and one that
       contains SAS code to correctly put, format and quote
       all variables into one long variable (for lines of
       data in the CSV. */
    data work._tempstc_statements;
        retain _statement_put _statement_names;
        length _statement_put _statement_names $ 10000;

        set work._tempstc_contents end=last;

        /* Type 2 is character, requires quote(strip())
           Type 1 is numeric, requires strip(put(best12.)). */
        if type = 2 then _statement_put = cats(_statement_put, "quote(strip(" || strip(name) ||
        "))");
        if type = 1 then _statement_put = cats(_statement_put, "strip(put(" || strip(name) || ",
        best12.))");
        if ^last then _statement_put = cats(_statement_put, "|"||"&delim."||");

        if last then call symputx("_statement_put", _statement_put);

        _statement_names = catx("&delim.", _statement_names, name);

```

```

        if last then call symputx("_statement_names", _statement_names);

        keep _statement_put _statement_names;
run;

%put &_statement_put;
%put &_statement_names;

%let option_quotelenmax=%sysfunc(getoption(quotelenmax));
%let option_missing=%sysfunc(getoption(missing));
option missing="";
option noquotelenmax;

/* Write the CSV file. */
data _null_;
    length line $ 32767;
    set &inds.;
    file "&outcsv." lrecl=32767;

    if _n_ = 1 then do;
        line = "&_statement_names.";
        put line;
    end;

    line = &_statement_put.;
    put line;

run;

option missing="&option_missing.";
option &option_quotelenmax.;

proc datasets lib=work nolist;
    delete _tempstc_;;
quit;run;

%put =====;
%put End of macro sasToCSV;
%put =====;

%mend sasToCSV;

```