

```

/*****
/* PROD PROGRAM:  ./MultiChkFMT.sas
/* WORK PROGRAM:  ./MultiChkFMT.sas
/*
/* PURPOSE:      The %MultiChkFMT macro scans through the defined format
/*                catalog(s) for overlapping format definitions and highlights
/*                which one will be used
/*
/* SOURCE PRGM:   none
/* INPUT:         NA
/* OUTPUT:        NA
/* MACROS USED:   none
/* EXEMPTIONS:    none
/*
/* AUTHOR:        Jeffrey Lavenberg
/* CREATION DATE: 12/03/2015
/*
/* NOTES:        The fmtsearch option, overrides the default value (WORK) and
/*                defines the specific libraries and the hierarchy SAS should
/*                follow
/*                Depending on the level of detail (submitted as a parameter)
/*                SAS will print out different information
/*                1 - Overlapping format names and order of definitions SAS will use
/*                2 - Format definitions SAS is using for overlapping formats names
/*                3 - Output from 1 and 2
/* MODIFICATIONS: none
*****/

```

```

%macro MultiChkFMT(detail);
  /* Get the list of libref specified in the fmtsearch option, removes parentheses from string */
  %let fmtsearch_locations=%sysfunc(tranwrd(%sysfunc(upcase(%sysfunc(compress(%sysfunc(getoption(fmtsearch)),"()")))),LIBRARY,));

  /* For each libref listed in the fmtsearch option */
  %do i=1 %to %sysfunc(countw(&fmtsearch_locations));
    /* Check if libref contains any formats */
    proc sql noprint;
      select count(*)
      from dictionary.formats
      having upcase(libname)=%sysfunc(quote(%sysfunc(upcase(%sysfunc(scan(&fmtsearch_locations,&i))))))
    ;
  quit;

  /* If the PROC SQL code came up empty, display warning */

```

```

%if &sqlobs=0 %then %put WARNING: Unable to locate any format(s) in: %sysfunc(scan(&fmtsearch_locations,&i));
%else %do;
    /* Output the format definitions found within the specific format catalog */
    proc format library=%sysfunc(scan(&fmtsearch_locations,&i)) cntlout=__fmt_%sysfunc(scan(&fmtsearch_locations,&i));
    run;

    /* Sort the dataset by format name */
    proc sort data=__fmt_%sysfunc(scan(&fmtsearch_locations,&i));
        by fmtname;
    run;

    data __fmt_%sysfunc(scan(&fmtsearch_locations,&i));
        length rank_lib $200;
        set __fmt_%sysfunc(scan(&fmtsearch_locations,&i));
        by fmtname;
        /* Create a variable with the number in which the libref was specified in the fmtsearch option */
        ranking=&i;
        /* Create a variable with the name of the libref */
        rank_lib=%sysfunc(quote(%sysfunc(upcase(%sysfunc(scan(&fmtsearch_locations,&i))))));

        /* Put nicer labels on the variables */
        label
            fmtname="Format Name"
            rank_lib="Library Name";

        /* Only print out the first record for each format definition */
        if first.fmtname;

        /* Only keep variables we are interested in */
        keep fmtname ranking rank_lib;
    run;
%end;

%if &i=%sysfunc(countw(&fmtsearch_locations)) %then %do;
    /* Create macro variable with values of the dataset names that contain at least one format definition */
    proc sql noprint;
        select memname into :mem_names separated by ' '
        from dictionary.tables
        where index(memname,"__FMT_") ge 1
        ;
    quit;

    /* If any datasets contain at least one format definition, set them all together */

```

```

%if %symexist(mem_names) %then %do;
    data __fmt_combine;
        set &mem_names;
run;

/* Create a dataset with all of the overlapping format names/definitions */
proc sql;
    create table __fmt_overlap as
    select fmtname, rank_lib
    from __fmt_combine
    group by fmtname
    having count(fmtname) ge 2
    order by fmtname, ranking
    ;
quit;

/* If the PROC SQL code came up empty, display note */
%if &sqlobs=0 %then %put NOTE: No overlapping format definitions found.;
%else %do;
    /* Print findings if detail equals 1 or 3 */
    title1 "Output from MultiChkFMT macro";
    title2 "Overview of overlapping format definitions";
    %if &detail = 1 or &detail = 3 %then %do;
        proc print data=__fmt_overlap label;
            by fmtname;
            id fmtname;
        run;
    %end;

    %if &detail = 2 or &detail = 3 %then %do;
        data _null_;
            set __fmt_overlap;
            by fmtname;

            if first.fmtname=1 then do;
                /* Loop through and print findings if detail equals 2 or 3 */
                call execute('title1 "Output from MultiChkFMT macro";');
                call execute('title2 "Details of overlapping format definitions being applied to datasets";');
                call execute("proc format library=" || rank_lib || ";");
                call execute("select " || fmtname || ";");
                call execute("run;");
            end;
        %end;
    %end;

```

```
                /* Clean up title statements */
                title1; title2;
            %end;
        %end;
    %end;
%end;

/* Remove temporary dataset */
proc datasets nolist;
    delete __fmt__;
run; quit;

/* Remove macro variable */
%symdel mem_names /nowarn;
%mend MultiChkFMT;

/* Example calls */
%MultiChkFMT(1);
%MultiChkFMT(2);
%MultiChkFMT(3);
```