

Creating Proc Format Code from Format Catalogs

Peter Dueber

Fred Hutchinson Cancer Research

Introduction

There are times when the code which generated a SAS format catalog is lost, is accidentally written over (with inadequate or no backup) or was simply never supplied along with the catalog file. It is not obvious how one could find out all the possible ranges and their labels or to modify a format hidden within the catalog file.

Here is a VERY simple macro to RECREATE THE EFFECTIVE PROC FORMAT CODE which originally generated the formats directly from the format catalog.

The Method

The first step is to get the format information into a dataset using the CNTLOUT option of the format procedure:

```
PROC FORMAT
  LIBRARY=arg1
  CNTLOUT=_fmts(KEEP=
    fmtname start end type label);
```

It is then a simple matter of inspecting the start, end, type and label variables to construct the code with the use of put statements. The resulting code includes an optional output library and can also be directed to a text file using a fileref:

```
PROC FORMAT <LIBRARY=arg2>;
  VALUE yesnomn
    1 = 'Yes'
    2 = 'No'
    3 = 'Maybe'
    4 = 'Never'
    .N = 'Not Available'
  ;

  PICTURE numpic
    0001 - 9999 = '999-9'
  ;

  ... etc.
```

The Example

For example, to access the formats contained in libname "fmtlib" and to create a new file called "projfmts.sas" that would, if run, create temporary work formats one would use the following statements:

```
LIBNAME  fmtlib  '/project/formats';
FILENAME fmtfile 'projfmts.sas';
%getfmt( inlib=fmtlib,
         fileref=fmtfile);
```

The Macro

```
%macro getfmt(inlib=WORK,
              outlib=WORK,
              fileref=LOG,
              faccess=NEW );

%*****;
%*  ARGS:                                     *;
%*  inlib-Format catalog libref *;
%*          to read(default=WORK) *;
%*                                     *;
%*  outlib-Destination libref   *;
%*          (default=WORK)      *;
%*                                     *;
%*  fileref-Fileref for the SAS *;
%*          code (default=LOG)  *;
%*                                     *;
%*  faccess-File access type    *;
%*          (for fileref):      *;
%*          MOD => append       *;
%*          NEW  => create      *;
%*****;

%* Get format info to a dataset *;
proc format
  library=&inlib
  cntlout=_fmts(keep=
    fmtname start end type label)
  ;

%* Partially construct the code *;
%* but save the rest for a      *;
%* subsequent data step to make *;
%* it look pretty               *;
proc sort data=_fmts;
  by fmtname start;
```

Creating Proc Format Code from Format Catalogs

Peter Dueber

Fred Hutchinson Cancer Research

```

data _fmts;
  length valpic $17 lout $60;
  set _fmts; by fmtname;

  start=trim(left(start));
  end=trim(left(end));

  /* Strip "*" chars;
  if start='**OTHER**' then
    start='Other';
  if end='**OTHER**' then
    end='Other';

  /* Format type
  if type='P' then
    valpic="PICTURE " ||trim(fmtname);
  else if type='C' then
    valpic="VALUE $" ||trim(fmtname);
  else if type='N' then
    valpic="VALUE " ||trim(fmtname);

  /* Char types need quotes;
  if type='C' then do;
    /* Ranges need a dash;
    if start^=end then
      lout=""" ||trim(start)||
        ""-"" || "" ||
        trim(end) ||""";
    else
      lout=""" ||
        trim(start) ||""";
  end;
  else do;
    if start^=end then
      lout=trim(start) ||"-||end;
    else
      lout=start;
  end;

  /* Get length of start-end;
  rlen=length(lout);
run;

/* Get max length of rlen so that;
/* we can line everything up ;
proc summary noprint nway;
  var rlen;
  class fmtname;
  output out=maxlen max=rmax;

  /* Insert rmax:create the code;
data _null_;
  merge _fmts maxlen;
  by fmtname;

  %if &fileref^=LOG %then
    %str(FILE &fileref &faccess;);

  if _n_=1 then
    put / "PROC FORMAT
      LIBRARY=&outlib;";

  /* value/picture statement;
  if first.fmtname then
    put / @3 valpic;

  /* start-end, hold the record;
  put @7 lout @;

  /* Line up the columns;
  do while (rlen<rmax);
    put " " @;
    rlen+1;
  end;

  /* Complete start-end = label;
  lout="= "" ||trim(label) ||""";
  put lout;

  * Finally, the semi-colon;
  if last.fmtname then
    put @7 ";";
run;
%mend getfmt;

```