IN & OUT of CNTL with PROC FORMAT

Nancy K. Patton, GE Capital

ABSTRACT (CNTLIN=)

If you were given a list of account numbers in a data set and asked to find all those accounts in numerous flat files, how might you approach this?

Using the SAS® system and the CNTLIN= option in PROC FORMAT, a format can be defined where all the account numbers you want identified are set to a value of "KEEP". By using the PUT function to apply that format to the account numbers as you read the flat files, you could identify the records with the accounts you are looking for and control their destiny!

This example will make the look-up process simple to understand and easy to bring back to work!

HOW TO: PROC FORMAT CNTLIN=

First, define the account numbers in a SAS data set, and create the variables which PROC FORMAT needs in order to produce a format using CNTLIN=. In this example you only need to use the variables: START, LABEL and FMTNAME. Refer to the documentation for other variables.

```
DATA LOOKUP;
INFILE DATAIN;
INPUT
  @20 IDNUMBER      $CHAR16. ;

START = IDNUMBER;
LABEL = 'KEEP';
FMTNAME = '$LOOKUP';
```

The CNTLIN= option of PROC FORMAT will not allow duplicates in the start variable, so using PROC SORT with the NODUPKEY option will ensure there are no duplicates.

```
PROC SORT
   DATA = LOOKUP
    OUT = FMT
   NODUPKEY;
BY START;
```

Finally, PROC FORMAT uses the data set we just created to create the format $LOOKUP.

```
PROC FORMAT CNTLIN=FMT;
```

You have a format defined as if you had typed the following format.

```
PROC FORMAT;
VALUE $LOOKUP
      '5326673453'='KEEP'
      '3857629859'='KEEP'
(… more account numbers …)
      ;
```

It is quite straightforward to use a DATA step to read the account number from data files and use a sub-setting IF statement to select the records which match.

```
IF PUT(account,$LOOKUP.)= 'KEEP';
```

Using the CNTLIN= option of PROC FORMAT gives you the control to define your own formats based on any data values available to your program, and minimized the possibilities of typing errors in data entry.

ABSTRACT (CNTLOUT=)

Have you ever hard-coded a 'dummy' data set to combine with your data in order to force all categories of a formatted CLASS variable in PROC SUMMAY be included in your output? You could have used CNTLOUT= option and created that data set quite easily!

You'll use this technique whenever you need to create a 'dummy' data set to force uniform output from PROC SUMMARY.

HOW TO: PROC FORMAT CNTLOUT=

The CNTLOUT= option needs to be added to the PROC FORMAT when creating the format to be used on the CLASS variable. This will create a CNTLOUT data set.

```
PROC FORMAT CNTLOUT=FMTOUT;
VALUE DISTRIB
  0-99 = 0
  100-199 = 100
  200-299 = 200
  300-399 = 300
  400-499 = 400
  500-599 = 500
  600-699 = 600
  700-799 = 700
  800-899 = 800
  900-999 = 900;
```

The data set FMTOUT consists of variables that give information about each format and informat created. We need to pay attention to the variable FMTNAME to select those variables in only the DISTRIB format. The LABEL variable will hold the actual value used to format the CLASS variables so it will be renamed to match. The other variables are documented and not necessary for this example.

```
%MACRO FILLDATA;
        DATA FILL;
          SET FMTOUT;
        IF FMTNAME = 'DISTRIB';
        KEEP LABEL;
        RENAME LABEL = CLSSVAR;
%MEND FILLDATA;
```

After calling and executing the macro FILLDATA, a SAS data set FILL is created. This can be appended to your data to force all classifications in the format to appear in the output of your PROC SUMMARY.

```
        %FILLDATA;

        DATA DIST1;
        SET SUGIDATA.DATA FILL;

        CLSSVAR =
        PUT (INPUT (GMSCORE), DISTRIB.);
```

```
        PROC SUMMARY
           NWAY MISSING DATA = DIST1;
           CLASS CLSSVAR;
          VAR AMOUNT;
        OUTPUT SUM= OUT=SUMM;
```

The data set DIST1 is made up of the raw data SUGIDATA.DATA and the data set FILL which you just created. After PROC SUMMARY is run on DIST1, the classification variable CLSSVAR will have each value in the format DISTRIB.

CONCLUSION

PROC FORMAT is an invaluable tool to use for reporting. PROC FORMAT in conjunction with CNTLIN= or CNTLOUT= gives you the ability to build formats from data and to use the formats to force classification variables to have values for all values of the format whether there is data or not.

The author can be contacted at:

Nancy Patton
GE Capital Corp.
805 S. Courthouse Rd.
Arlington, VA 22204-2104
(703) 892-4472
Nancy.Patton@GECapital.com