

Paper BB-113

Formatting Data in SAS – Easy and Powerful

Leanne Tang, National Agricultural Statistics Service USDA, Washington DC

ABSTRACT

In our organization a lot of efforts are put into building and maintaining our organizational level metadata databases. Metadata databases are used to store information, or metadata, about our data. Many times we have to “decipher” the keys and codes associate with our data to enhance our data presentation. One of the options to interpret our data is to generate user defined formats from metadata using SAS® PROC FORMAT. In this paper we are going to explore some of the powerful options available in SAS® PROC FORMAT procedure. We will create user defined formats from metadata. We will look into different ways using the formats.

OVERVIEW

The SAS® System provides many pre-build informats and formats for formatting character, number, and date. Informats are instructions for reading data. Formats are used to format output. Many times pre-build formats cannot meet the needs of our data use. So we have to create our own formats - user defined formats using SAS® PROC FORMAT procedure. Here is the basic syntax for PROC FORMAT from support.sas.com:

```
PROC FORMAT <option(s)>;
  EXCLUDE entry(s);
  INVALUE <$>name <(informat-option(s))>
    value-range-set(s);
  PICTURE name <(format-option(s))>
    value-range-set-1 <(picture-1-option(s) )>
    <...value-range-set-n <(picture-n-option(s))>>;
  SELECT entry(s);
  VALUE <$>name <(format-option(s))>
    value-range-set(s);
```

WHAT ARE CNTLIN AND CNTLOUT?

In this paper we are going to focus on two powerful PROC FORMAT options - CNTLIN and CNTLOUT.

CNTLIN option can be used to specify a SAS data set for building informats and formats with PROC FORMAT procedure. This opens a door to endless possibilities for creating user defined formats.

CNTLOUT option can be used to generate SAS data set with information about formats and informats. This is especially useful when we inherit permanent formats and like to learn more about the formats.

CREATING FORMATS FROM METADATA DATABASE

There are three simple steps to create basic formats from metadata database using PROC FORMAT.

Step 1. Connect to Database

First we create libref, myDatabase, by using LIBNAME statement. We are using ODBC Driver to connect to metadata database:

```
LIBNAME myDatabase ODBC USER=tangle PW=XXXXX DSN=myDSN CONNECTION=UNIQUE;
```

A link to Microsoft.com is available in reference section on how to create ODBC driver.

Step 2. Get Metadata

Next we create a SAS data set with information we need to create formats from metadata database. In this example we are going to get data for formatting farm IDs to farm names.

```
DATA farmListFmts (KEEP=fmtName start label);
  SET farmList ;
  RETAIN fmtname 'farmName';
  start= farmID;
  label= farmName;
RUN ;
```

There are three required columns for PROC FORMAT procedure, fmtname, start, and label. FMTNAME column is used to store user defined format names. In this example we name our formats 'farmName'. START column should contain data need to be formatted. LABEL column should have data we are going to format to.

Output 1. farmListFmts SAS Data Set

	fmtname	start	label
1	farmName	101	Alabama Cotton Farm
2	farmName	102	Alabama Poultry Farm
3	farmName	103	Alabama Livestock Farm
4	farmName	601	California Horticulture Farm
5	farmName	602	California Fruit Tree Farm
6	farmName	481	Texas Livestock Farm
7	farmName	482	Texas Horticulture Farm
8	farmName	483	Texas Equestrian Farm

In this example we are formatting the farm ID which is numeric data type. SAS default format data type is numeric. We can specify the type explicitly by creating the 'type' in the procedure:

```
DATA farmListFmts (KEEP=fmtName start label);
  SET farmList ;
  RETAIN fmtname 'farmName' type 'N' ;
  start= farmID;
  label= farmName;
RUN ;
```

Perhaps we need to create formats for range of data. When formats are applied the ranges are searched for the data to be formatted. This can be accomplished by adding 'end' column in the table.

The table below shows metadata for strata with total sale range. We are going to use 'low' and 'high' for 'start' and 'end' to create range formats.

Output 2. strataFmts SAS Data Set

	fmtname	start	end	label
1	strata	0	99.99	1
2	strata	100	999.99	2
3	strata	1000	4999.99	3
4	strata	5000	9999.99	4
5	strata	10000	100000	5

By using 'start' and 'end' key columns we instruct SAS to look for data between the ranges and apply strata formats.

```

DATA strataFmts (KEEP=fmtName start end label);
  SET strata ;
  RETAIN fmtname 'strata';
  start= low ;
  end=high ;
  label= stratum;
RUN ;

```

Step 3. Creating Formats

Now data in farmListFmts and strataFmts SAS data sets are ready to be used to create formats. After executing following PROC FORMAT procedures user defined farmName and strata formats will be available in work.formats catalog.

```

PROC FORMAT CNTLIN=farmListFmts ;
PROC FORMAT CNTLIN= strataFmts;
RUN ;

```

USING FORMATS

There are many different ways to use formats other than formatting data. Here are three examples to apply formats we had just created above.

1. Formatting Data In PROC PRINT Procedure

Most commonly we format data in PROC PRINT to enhance data presentation. The user defined formats can be applied like any SAS® System Formats. In this procedure we use farmName formats to format farm IDs to farm names:

```

PROC PRINT DATA=farmData LABEL;
  FORMAT farmID farmName. ;
  LABEL farmID ='Farm Name' totalSale = 'Total Sale';
RUN ;

```

Output 3. Data Before and After Formats Were Applied

	FarmID	totalSale
1	101	500
2	102	0
3	102	0
4	103	1200
5	103	1200
6	481	5200
7	482	4000
8	601	200
9	602	200000

Obs	Farm Name	Total Sale
1	Alabama Cotton Farm	500
2	Alabama Poultry Farm	0
3	Alabama Poultry Farm	0
4	Alabama Livestock Farm	1200
5	Alabama Livestock Farm	1200
6	Texas Livestock Farm	5200
7	Texas Horticulture Farm	4000
8	Califonia Horticulture Farm	200
9	Califonia Fruit Tree Farm	200000

2. Creating New Data Values

Formats can also be used to generate new data values. In the example below we create two new columns, farmName and stratum, by applying farmName formats to farm IDs and strata formats to totalSale.

```

DATA dataFmt ;
  SET DataNoFmt ;
  farmName = PUT(farmID, farmName.);
  stratum = PUT(totalSale, strata.);
RUN ;

```

Output 4. New farmName and Stratum Columns

	FamID	totalSale	famName	stratum
1	101	500	Alabama Cotton Farm	2
2	102	0	Alabama Poultry Farm	1
3	102	0	Alabama Poultry Farm	1
4	103	1200	Alabama Livestock Farm	3
5	103	1200	Alabama Livestock Farm	3
6	481	5200	Texas Livestock Farm	4
7	482	4000	Texas Horticulture Farm	3
8	601	200	California Horticulture Farm	2
9	602	200000	California Fruit Tree Farm	*

3. Data Lookup in WHERE Statement

Formats can also be used in WHERE statement to subset data. In this example we create a new SAS data set with only livestock farms by using farmName formats in the WHERE statement.

```
DATA livestockFmtNameStrata ;
    SET DataFmtNameStrata ;
    WHERE PUT(farmID, farmName.) LIKE '%Livestock%';
RUN ;
```

Output 5. Livestock Farm Data

	FamID	totalSale	famName	stratum
1	103	1200	Alabama Livestock Farm	3
2	103	1200	Alabama Livestock Farm	3
3	481	5200	Texas Livestock Farm	4

TEMPORARY FORMATS VS PERMANENT FORMATS

So far we discussed about user defined formats created in work.formats catalog. They are temporary formats. We lose them when we exit current SAS session. Because they can be created right before we need them we are getting the most up-to-date metadata. For metadata gets updated frequently creating temporary formats can be a very suitable practice.

When metadata are relatively stable we can save our execution time by creating and storing permanent formats ahead of time. To create permanent formats, first specify the location in LIBNAME statement. Then, in PROC FORMAT, include LIBRARY option and specify the library or catalog to have the formats to be saved to.

```
LIBNAME FmtsLib 'O:\EES\SAS User Groups\SESUG2014' ;
PROC FORMAT CNTLIN=farmListFmts LIBRARY = FmtsLib ;
PROC FORMAT CNTLIN=strataFmts LIBRARY = FmtsLib ;
RUN ;
```

When you have formats in multiple libraries or catalogs specify your search order to avoid default SAS search order. By default, SAS search WORK library first then user created libraries. To avoid default search order you make sure to specify your preferred search order before applying the formats.

```
options fmtsearch=(FmtsLib work);
```

CONCLUSION

In our organization more and more metadata are centralized and accessible to us. To improve our productivity we need to make sure we take advantage of the information available to us. The powerful PROC FORMAT can turn

pretty much any metadata into formats. Formats can not only be used to format data but can be used to create new data values, subset data, validate data, etc. We only touched a fraction of what PROC FORMAT can do. To learn more about PROC FORMAT procedure and its options go to SAS website.

The SAS programs in this paper are tested using SAS® 9.2.

REFERENCES

SAS Online Procedure Documents. Available at

<http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a000063536.htm>

SAS Formats and the Format Procedure – Course Notes. Available at

http://www.wuss.org/proceedings11/Papers_Riddiough_C_77625.pdf

Creating a Format from Raw Data or a SAS® Dataset. Available at

<http://www2.sas.com/proceedings/forum2007/068-2007.pdf>

How Do We CNTL It? Available at <http://www.lexjansen.com/nesug/nesug01/cc/cc4021.pdf>

New Tricks for an Old Tool: Using Custom Formats for Data Validation and Program Efficiency. Available at

http://www.jadetek.com/download/jade_fmt.pdf

Some Useful Techniques of Proc Format. Available at <http://www.pharmasug.org/proceedings/2011/CC/PharmaSUG-2011-CC19.pdf>

Ten Things You Should Know About PROC FORMAT. Available at

<http://www.ats.ucla.edu/stat/sas/library/nesug00/bt3014.pdf>

How to: Set Up an ODBC Data Source. Available at [http://msdn.microsoft.com/en-us/library/ca6axakh\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/ca6axakh(v=vs.80).aspx)

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Leanne Tang
National Agricultural Statistics Services, USDA
1400 Independence Ave., SW
Washington, D.C 20250-2001
202 – 720 6957
Leanne_tang@nass.usda.gov

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.