Paper TU03

PICTURE Perfect: In depth look at the PICTURE format.

Carry W. Croghan, US-EPA, Research Triangle Park, NC.

ABSTRACT

SAS PICTURE format is a very powerful tool. The different options allow the display of data in a more picturesque manner adding symbols, qualifiers, and comments to the data points without modifying the data. With a PICTURE format one can create a series of templates for displaying the data. The PICTURE format even allows the user to generate their own date and time formats. In addition to the options available with the VALUE formats, PICTURE format has options such as DATATYPE, NOEDIT, PREFIX, MULTIPLIER, and FILL. With its multitude of options, the PICTURE format is very impressive, but it can also be confusing. It is not always clear how different options will interact. If not carefully applied, the PICTURE format can produce misleading information. This presentation will look at the structure and the options available, removing some of the mystery of the PICTURE format and revealing a powerful formatting tool.

INTRODUCTION

Formats allow you to display the data in a different structure without modifying the data. There are two general types of formats that a user can define:

1. VALUE - assigns a label or text string to a range of values,
2. PICTURE - creates templates in which you define how the numbers are displayed. With a PICTURE format you can output text strings as well as the numeric values.

PROC FORMAT SYNTAX

To define a format you use the PROC FORMAT procedure. The basic syntax is

```
proc format ;
     picture name <(options)>
          range1 = 'Template1' <(options)>
         <range2 = 'template2' <(options)>
             . . .
          rangek = 'templatek' <(options)> >;
```

A name is required. It is an alphanumeric string of length up to 8 with the additional restriction that it cannot end with a number. Also, it cannot be a name of an existing SAS defined format.

A range can be a single value (0), a set of values separated by commas (i.e. 2,4,6), or an interval of values (i.e. 1-6). There are 3 key words that can be used in defining the range: Low – smallest non-missing value, High – largest value, and Other – all values not in any of the ranges. To define an exclusive range, the symbol '<' is used. This can be used at either the beginning or end of the range. For example, low -<0 would include all the negative values and would not include 0.

A template consists of digit selectors and/or sequence of characters within quotes. The template must start with a digit selector. The digits are interpreted as numeric place holders in which the data are placed. A zero is a place holder that does not print leading zeros. All other digits (1-9) are place holders that do print leading zeros. It is tradition to only use 0 and 9 in the templates. This is a convention that all the examples will follow. There is a limit of 16 digit selectors in a template. Most characters are message characters. They will be displayed exactly as you have written them. There are some special characters combinations that are directives used in the creation of date and datetime templates. These will be discussed in detail later on.

This is a simple example of a PICTURE format for checking a validity of a month variable.

```
proc format;
     picture mon
            1-12 ='99'
            other = 'Not a valid month';
run;
```

There are two range specifications and two templates. The '99' in the first template dictate that a leading zero is printed. The second template is used for all values outside of range1. This template is a text message.

Here is how some values would be displayed.

Original data	Formatted data
0	Not a valid month
1	01
2	02
13	Not a valid month
10	10

Changing the format slightly, the leading zeros can be removed by using 0 as the digit selector.

```
proc format;
     picture mon
            1-12 ='00'
            other = 'Not a valid month';
run;
```

Original data	Formatted data
0	Not a valid month
1	1
2	2
13	Not a valid month
10	10

What happens if the "other" range is removed?

Original data	Formatted data
0	0
1	1
2	2
13	13
10	10

This is not a very effective format. The formatted data look no different from the unformatted data. The other range is an important part of the format. Another way to format would be to use a VALUE format for the non-valid month values.

```
proc format;
     value mon
            low-<1,12<-high = 'Not a valid month';
run;
```

However, what a PICTURE format can do but a VALUE format cannot is combine the data with text.

```
proc format;
     picture mon
            1-12 ='99'
        other ='09 is not valid';
run;
```

Original data	Formatted data
0	0 is not valid

Original data	Formatted data
1	01
2	02
13	13 is not valid
10	10

This format works fine for integer values. However, it would not work for non-integers.

To paraphrase the ancient Delphi Oracle, 'Know thy data'. This is especial important when working with PICTURE formats. You need to know the magnitude and the precision of the data values. The following example uses the same format with different data values.

Original data	Formatted data
-11.0	11 is not valid
1.5	01
30012.0	12 is not valid

Although none of the values are valid months, 1.5 is reported as valid, and for the others the print out is misleading.

The PICTURE format not only truncates the data to the right of the decimal, it truncates the data to left of the defined template. It also takes the absolute value of the data.

It is important to check the data to verify the range and precision. The simplest method is to do a PROC FREQ. If it is a large dataset with numerous distinct values which should be whole numbers, create an indicator variable to check if the data are in fact whole numbers.

```
chk = (round(X)=X);
```

Then run a PROC MEANS with MIN and MAX options on the original variable X and the new indicator variable chk. Taking a few minutes to check the data can save hours of later head scratching.

To understand how SAS applies a template, we need to examine the specific steps that SAS goes through. These steps can be modified by many of the options; therefore the options are defined first and then the steps.

OPTIONS

There are several options available in PROC FORMAT. These options are divided into two general types:

1. Overall options - apply to the entire format.
2. Template options - apply to a specific range and template.

Most of the overall options can be used with VALUE as well as PICTURE formats.

Option	Description
DEFAULT=	Specifies a default length for the format.
FUZZ=	Specifies a fuzz factor for matching values to a range.
MAX=	Specifies a maximum length for the format
MIN=	Specifies a minimum length for the format
MULTLABEL	Allows your ranges to overlap. This option is only useful if the called procedure allows multiple labels.
NOTSORTED	Keeps the data in the order that you defined.
ROUND	Round the data to the nearest integer before formatting.

These template options are applicable only to PICTURE formats.

Option	Description
DATATYPE=	Specifies the type of format. Only needed if you are defining a date, time, or datetime format. The only valid values are DATE, TIME, and DATETIME.
DECSEP	Specifies a decimal separator character; by default it is '.'
DIG3SEP	Specifies the three digit separator character; by default it is ','
FILL=	Specifies a value to fill in the blank space.
MULTIPLIER=, MULT=	Specifies a value to multiply the data by before it is formatted.
NOEDIT	The template is printed as is, the numeric values are interpreted as part of the text and data is not displayed.
PREFIX=	Specifies a string of characters to place before the formatted values.

HOW SAS APPLIES A TEMPLATE

In the first step, SAS locates the range that the data value is in. If you have used the MULTLABEL option, it locates the primary range. What the primary range is depends on whether the NOTSORTED option is used. Without the NOTSORTED option, the primary range is the one that is numerically first. To determine which range is numerically first both the start and the end of the range is considered. The range 0-9 occurs before 0-99. The range 0-9999 occurs before 10-999. With the NOTSORTED option, it is the first range that you define in which the data value is in. The FUZZ factor is also applied at this step. The FUZZ factor is similar to rounding; however, it is only used to match a data value to a range.

In the second step, SAS takes the absolute value of the data. SAS applies the multiplier in the third step. This value is either implicit or explicit. If you used the MULT option, then the multiplier is explicitly defined. The implicit multiplier is based on the template. It is 10n, where n is the number place to the right of the decimal indicator. (The decimal indicator is usually '.'. You can used the DECSEP option to change it to another character.)

In the fourth step, unless the ROUND option is used the data value is truncated to fit in the template. If the ROUND option is used the data value is rounded to the nearest integer value.

In the fifth step, the numeric data is converted to a character string. If the magnitude of the value is smaller than the number of digit selectors in the PICTURE, SAS pads the values to the left with leading zeros.

Next the new character string is placed in the template starting with the right most digit selector. If the first digit selector is a zero, the leading zeros are replaced with blanks or the alternative FILL value.

The final step is the placing of the prefix if the PREFIX option was used. If the template is not wide enough to accommodate the data and the prefix, SAS will drop the prefix.

The best way to explain both the steps and the options is by example.

Say we have the following recovery fractions data values: 1.22849, 2.09283, 0.64999,
-0.09871, and 0.23302. Let us assume that a recovery between 65% and 175% is considered to be good, any thing larger than 175% is considered to indicate contamination, smaller than 65% indicates a bad recovery. Since recoveries are calculated as absolute differences, a negative value indicates erroneous data. The data precision should only be at the 0.001 level. So we want the data rounded and the ranges to be assigned to the appropriate precision. The researchers prefer to look at the results as percent recovery.

The format statements would be.

```
proc format;
     picture recovery (round fuzz=0.005 default = 25)
            low-<0 = '999.9% Not valid Data' (prefix='-' mult = 1000)          /*1*/
            0-0.64 = '009.9% Bad recovery' (mult = 1000)                              /*2*/
            0.65  - 1.75 = '009.9%' (mult = 1000)                                     /*3*/
            1.75<-high = '00009.9%' (prefix = 'Contamination.  ' mult=1000)    /*4*/;
run;
```

STEP 1. SAS ASSIGNS A RANGE.

Original values	Range
1.22849	3
2.09283	4
0.64999	3 (FUZZ = 0.005)
-0.09871	1
0.23302	2

STEP 2. SAS TAKES THE ABSOLUTE VALUES.

Original values	Range	Absolute values
1.22849	3	1.22849
2.09283	4	2.09283
0.64999	3	0.64999
-0.09871	1	0.09871
0.23302	2	0.23302

STEP 3. SAS MULTIPLIES BY THE MULT VALUE.

Original values	Range	Absolute values	MULT
1.22849	3	1.22849	1228.49
2.09283	4	2.09283	2092.83
0.64999	3	0.64999	649.99
-0.09871	1	0.09871	98.71
0.23302	2	0.23302	233.02

STEP 4. SAS ROUNDS OR TRUNCATIONS.

Original values	Range	Absolute values	MULT	ROUND
1.22849	3	1.22849	1228.49	1228
2.09283	4	2.09283	2092.83	2093
0.64999	3	0.64999	649.99	650
-0.09871	1	0.09871	98.71	99
0.23302	2	0.23302	233.02	233

STEP 5. SAS CONVERTS TO CHARACTER AND PADS WITH LEADING ZEROS

Original values	Range	Absolute values	MULT	ROUND	Char
1.22849	3	1.22849	1228.49	1228	1228
2.09283	4	2.09283	2092.83	2093	2093
0.64999	3	0.64999	649.99	650	0650
-0.09871	1	0.09871	98.71	99	0099
0.23302	2	0.23302	233.02	233	0233

STEP 6. SAS REPLACES DIGIT SELECTORS IN THE TEMPLATE, KEEPING ONLY THE SPECIFIED LEADING ZEROS.

Original values	Range	Char.	Templates	Insert Data
1.22849	3	1228	'009.9%'	122.8%
2.09283	4	2093	'00009.9%'	209.3%
0.64999	3	0650	'009.9%'	65.0%
-0.09871	1	0099	'999.9% Not valid Data'	009.9% Not valid Data
0.23302	2	0233	'009.9% Bad recovery'	23.3% Bad recovery

STEP 7. SAS ADDS THE PREFIXES.

Original values	Range	Insert Data	Add Prefixes
1.22849	3	122.8%	122.8%
2.09283	4	209.3%	Contamination. 209.3%
0.64999	3	65.0%	65.0%
-0.09871	1	009.9% Not valid Data	-009.9% Not valid Data
0.23302	2	23.3% Bad recovery	23.3% Bad recovery

EXAMPLES OF SELECT OPTIONS

PREFIX AND ROUND

Since a template must start with a digit selector, the only way to get text in front of the data is to use the PREFIX option. Here is an example of the PICTURE format using both the ROUND and PREFIX options. In this example we will be formatting data for an accounting report. The position values will be reported in dollars and the negative values are in ().

```
proc format;
     picture accA (round)
            low-<0  = '0,000,009)'        (prefix = '(')
            0-high  = '0,000,009' (prefix = '$');
run;
```

Original data	Formatted data
-980.00	(980)
-745.60	(746)
200.00	$200
348.75	$349
880.20	$880
1420.00	$1,420
5000000.00	$5,000,000

If the ROUND option is omitted, the fractional portion of the data will be truncated.

```
proc format;
     picture accB /*No round option */
            low-<0  = '0,000,009)'        (prefix = '(')
            0-high  = '0,000,009' (prefix = '$');
run;
```

Original data	Formatted data
-980.00	(980)
-745.60	(745)
200.00	$200

Original data	Formatted data
348.75	$348
880.20	$880
1420.00	$1,420
5000000.00	$5,000,000

Too small of a template as well as not using the ROUND option cause the data to be truncated both on the right and left.

```
proc format;
        picture accB /*No round option */
                low-<0  = '009)'
(prefix = '(')
                0-high  = '009'
(prefix = '$');
run;
```

Original data	Formatted data
-980.00	980)
-745.60	745)
200.00	$200
348.75	$348
880.20	$880
1420.00	$420
5000000.00	$0

Notice the lose of not only data information but also the prefix on the negative values.

NOEDIT

Another option that I frequently use is the NOEDIT option. It instructs SAS to treat any numbers within the template as characters and not as digit selectors. For example,

```
proc format;
        picture accB        (round default=10
 fuzz= .5)
                low-<0  = '009)'    (prefix = '(')
                0-999  = '009'      (prefix = '$')
1000-high = '>      1K' (noedit);
run;
```

Original data	Formatted data
-980.00	(980)
-745.60	(746)
200.00	$200
348.75	$349
999.60	> 1K
1420.00	> 1K
5000000.00	> 1K

With DEFAULT option used, there is enough space for the '(' to be displayed for the negative values. The FUZZ factor causes 999.6 to be assigned to the third range. Without FUZZ, the value 999.6 is not within any of the ranges. (Remember, rounding is not done until step 4.) The NOEDIT indicates that the 1 in the last template should be printed as is and not replaced with data values. Without the NOEDIT option, the output is misleading.

Original data	Formatted data
-980.00	(980)
-745.60	(746)
200.00	$200
348.75	$349
999.60	0K
1420.00	0K
5000000.00	0K

CREATING YOUR OWN DATES FORMATS

SAS provides a plethora of date and time formats. If you want something more esoteric, you can create your own by including special characters in the PICTURE template called DIRECTIVES. DIRECTIVES are only recognized by SAS when you specify the option DATATYPE in the template. A DIRECTIVE is percent sign followed by a letter.

The permitted directives are

Directive	Description
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%d	Day of the month as a number (1-31)
%H	Hour (24-hour clock) as a number (0-23)
%I	Hour (12-hour clock) as a number (1-12)
%j	Day of the year as a number (1-366)
%m	Month as a number (1-12)
%M	Minute as a number (0-59)
%p	either AM or PM
%S	Second as a number (0-59)
%U	Week number of the year (0,53)
%w	Weekday as a number (1= Sunday, 7)
%y	Year without century as a number (0-99)
%Y	Year with century as a number
%%	%

By default none of the DIRECTIVES generate leading zeros. By adding a 0 between the % and the letter of a numeric directive, you can get leading zeros. For example, if you specify %0y, then for the year 2004 you will get 04.

With a DATATYPE PICTURE format, it is important to explicitly define the default length using the DEFAULT option. SAS counts the space between the quotes as the default length unless otherwise directed. Since %B is only two spaces it will never accommodate the full name of the month.

```
proc format ;
  picture mydate    /*No Default */
            low-high = '%B' (datatype=date);
run;
```

Original Data (formatted with date9.)	Formatted data
14JUN2004	Ju
29JUL2004	Ju

Original Data (formatted with date9.)	Formatted data
17SEP2004	Se

When choosing a DEFAULT, take into account the largest possible value.

```
proc format ;
  picture mydate   (default = 15)
          low-high = '%B' (datatype=date);
run;
```

Original Data (formatted with date9.)	Formatted data
14JUN2004	June
29JUL2004	July
17SEP2004	September

Here is a more elaborate date format.

```
proc format;
  picture mydate   (default = 30)
          low-'04NOV2004'd = '%a %d of %B %Y' (datatype=date)

          '04NOV2004'd - high = 'Sometime in the future';
run;
```

Original Data (formatted with date9.)	Formatted data
09JUL2004	Fri 9 of July 2004
23AUG2004	Mon 23 of August 2004
07OCT2004	Thu 7 of October 2004
21NOV2004	Sometime in the future

CONCLUSIONS

PICTURE formats allows the combination of text messages and data in a structured output. The steps that SAS goes through to input the data into the template are important to remember. There are many options that must be used to produce the desired output. Some important options are PREFIX, NOEDIT, ROUND, and DEFAULT. PREFIX allows the placement of text before the numbers. This is important because a template must start with a digit selector. Any text in front of the first digit selector is ignored by SAS. If any numbers are included in a text message, SAS will interpret those as digit selectors unless the option NOEDIT is used. By default SAS truncates the data. The ROUND option is important in order for the continuous data to not be truncated. Get all of the desired text output by explicitly defining the length of the format by using the DEFAULT option. However, the most important thing to remember is to "Know thy data". It is easy to produce misleading output with a PICTURE format if the template does not accommodate the data, or the specified ranges do not capture all the data values. When used with care, a PICTURE format is a powerful tool for displaying data.

REFERENCE

Lund, Pete, 2002 , "More than Just Value: A Look Into the Depths of PROC FORMAT". SUGI 27.

SAS, 1999, SAS Procedures Guide, Version 8. SAS Publishing.

CONTACT INFORMATION

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

Author Name: Carry Croghan
Company: US - EPA
Address: 109 TW Alexander Drive
City state ZIP: RTP, NC 27709

Work Phone: (919) 541 3184
Email: croghan.carry@epa.gov

This paper has been reviewed in accordance with the United States Environmental Protection Agency's peer and administrative review policies and approved for presentation and publication. Mention of trade names or commercial products does not constitute endorsement or recommendation by EPA for use.