

In last tutorial, you learned to associate SAS built-in formats with variables either temporarily or permanently using the FORMAT statement. But sometimes you might want to create custom formats (also called [user defined formats](#)) for displaying variable values. Using the [FORMAT procedure](#), you can define your own formats for variables.



The FORMAT procedure creates formats that will later be associated with variables in a FORMAT statement. The procedure starts with the statement PROC FORMAT and continues with one or more VALUE statements:

PROC FORMAT;

```
VALUE format-name      range-1 = 'formatted-text-1'  
                        range-2 = 'formatted-text-2'  
                        .....  
                        range-n = 'formatted-text-n';
```

where format-name

- must begin with a dollar sign (\$) if the format applies to character data
- must be a valid SAS name
- cannot be the name of an existing SAS format
- cannot end in a number
- does not end in a period when specified in a VALUE statement
- Beginning with SAS®9, a numeric format name can be up to 32 characters in length. A character format name can be up to 31 characters in length.



PROC FORMAT;

```
VALUE format-name      range-1 = 'formatted-text-1'  
                        range-2 = 'formatted-text-2'  
                        .....  
                        range-n = 'formatted-text-n';
```

Each **range**

- is the value of a variable that is assigned to the text given in quotation marks on the right side of the equal sign.
- Character values should be enclosed in quotation marks (e.g. 'A' = 'Asia').
- If there is more than one value in the range, then separate the values with a comma or use the hyphen (-) for a continuous range. (e.g. 1, 3, 5, 7, 9 = 'Odd')
- The keywords LOW and HIGH can be used in ranges to indicate the lowest and the highest non-missing value for the variable. (The keyword LOW does not include missing numeric values. BUT If applied to a character format, the keyword LOW includes missing character values.) (e.g. 500000 - HIGH = 'Not Affordable')
- You can also use the less than symbol (<) in ranges to exclude either end point of the range. (e.g. 13 -< 20 = 'Teenager')
- The OTHER keyword can be used to assign a format to any values not listed in the VALUE statement. (e.g. OTHER = 'Bad Data')



The following are examples of valid **range** specifications:

'A' = 'Asia'

1, 3, 5, 7, 9 = 'Odd'

500000 - HIGH = 'Not Affordable'

13 -< 20 = 'Teenager'

0 <- HIGH = 'Positive Non Zero'

OTHER = 'Bad Data'



Assigning Your User-defined Formats to Variables

Just as with SAS formats, you associate a user-defined format with a variable in a FORMAT statement. When associating a format with a variable, remember to

- use the same format name in the FORMAT statement that you specified in the VALUE statement
- place a period at the end of your user-defined format name when it is used in the FORMAT statement. (Note: a period is *not* required at the end of the SAS built-in formats)
- If you do not format all of a variable's values, then those that are not listed in the VALUE statement are printed as they appear in the SAS data set

Example:

```
PROC FORMAT;  
VALUE $genderf    'm' = 'Male'  
                  'f' = 'Female';  
  
PROC PRINT DATA = scoredata1;  
FORMAT gender $genderf. ;  
Run;
```

