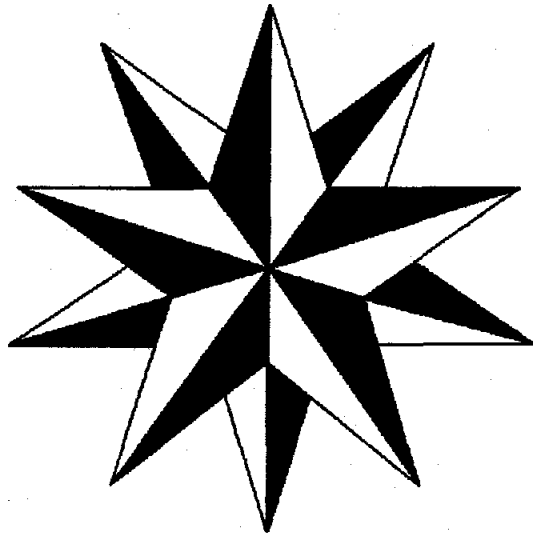


LIFE IN THE FAST LANE
USING PROC FORMAT



Kim Kolbe
Kohler Company

Using Proc Format to Save Time:

1. Min/Max Option
2. Fuzz Option
3. Utilizing Picture Statement
4. Utilizing Invalue Statement (available only in Versions 5.18 and up)
5. Table Lookups-- Efficient Programming/Recoding
6. Using Proc Format with Other Procs to Tailor Output
7. Using Proc Format to Avoid Hard-Coding AF screens
8. Permanent Format Library

Topics of Discussion:

1. The first step to saving time (both processing and programming) is to properly understand and utilize all the features and options that currently exist in Proc Format .
2. And then, when the option exists to solve a problem one of two ways, to utilize the Proc's strengths to solve the programming problem in the most efficient manner.

Proc Format

Internal

Format NetSales Dollar20.;

User Defined

```
Proc Format;  
  Value Statefmt  
    1='AL'  
    2='AZ'....;  
Format State Statefmt.;
```

How it is Used

What is the advantage of using Input vs. Informat?

Informat permanently associates the format with the variable, while input is only valid for the duration of processing.

Input	Input NetSales; Format NetSales 8.2; or Input NetSales 8.2;
Informat	Informat NetSales 8.2; Input NetSales;
Attribute	Attrib NetSales Length =8 Informat=Dollar10.;; Input NetSales;
Put	Put NetSales; Format NetSales Dollar10.;; or Put NetSales Dollar10.;; or Attrib NetSales Length=8 Format=Dollar10.;; Put NetSales;

Proc Format

Picture

- * Similar to internal formats such as Comma, Dollar, SSN
- * A user defined template for output
- * Numeric only
- * Limited to a length of 24

Value

- * What we are more accustomed to using when we think of Proc Format
- * Limited to a length of 16

What You Might Not Know :

Use of ranges

Value	(single value)
Value-Value	(range of values)
Range,Range	(list of ranges or values)

Be cautious of overlapping values!!

Value Agefmt	1-12	= 'Child'
	13-15,17-19	= 'Teen'
	16	= 'Sweet Sixteen'
	0-Low	= 'Miscoded'
	20-High	= 'Adult';

For 0-Low range **Other** could have also been used

If the value statement does not include all of a variable's values, those in the VALUE statement are printed with the specified format, while other values are printed with a default format.

Range specification above is applicable for both Value and Picture statements.

Min/Max Option

```
Proc Format;  
    Value $State (Min=3 Max=10)  
        1='Alabama' 2='Arkansas' 3='Arizona' 4='California' 5='Illinois';  
Data Test; Infile Cards;  
    Input @1 State $ 1. @3 Qty 3.;  
Cards;  
3 100  
2 389  
5 250  
;  
Proc Print Noobs; Format State $State.;  
Proc Print Noobs; Format State $State3.;
```

Produces:

State	Qty
Arizona	100
Arkansas	389
Illinois	250

State	Qty
Ari	100
Ark	389
Ill	250

Min/Max option controls length of the output format. Utilizing the Min/Max option on the Proc Format, can be particularly useful when attempting to prevent a report from wrapping. Length statements only control the variable length, not the format length.

Fuzz Option

```
Proc Format;  
  Value Gradefmt (Fuzz=.5)  
    4='A' 3='B' 2='C' 1='D' Other='Flunk';  
Data Test; Infile Cards;  
  Input @1 Grade 3.1;  
Cards;  
3.1  
1.9  
3.8  
4.1  
0.2  
;  
Proc Print Noobs; Format Grade Gradefmt.;
```

Produces:

```
Grade  
B  
C  
A  
A  
Flunk
```

Fuzz option saves programming time in creating format ranges. By using Fuzz=.5, this means if a number is within .5 of an input value, it is seen as a match.

Fuzz option only applies to Value statements, it is not available for Picture statements.

Picture Statement

```
Proc Format;  
  Picture Nsfmt  
    Other='09.9M'  
    (Prefix='$' Mult=.00001);  
Data Test; Infile Cards;  
  Input @1 Earnings 8.;  
Cards;  
100000  
1000000  
3200000  
;  
Proc Print Noobs; Format Earnings Nsfmt.;
```

Produces:

```
Earnings  
$0.1M  
$1.0M  
$3.2M
```

The picture statement provides additional flexibility in formatting output. It is useful in "tailoring" output with special characters. Some options available on the picture statement are:

- | | |
|--------------------|--|
| Fill= ' ' | Allows the field to be left-filled with a specified character |
| Prefix= ' ' | Allows one or two characters to precede the number. It occupies a position within the picture, it is not in addition to the variable's length. If there is not enough room, it will truncate or omit data. |
| Mult= | Allows output to be scaled either downward or upward.
I.e. Mult=.0001 scales a number downward, while
Mult=1000 would scale a number upward. |

Use picture options with caution, if your variable field length is not long enough, results can sometimes be unpredictable.

Picture Statement

Another Example:

```
Proc Format;
  Picture Phonefmt
    1000000-9999999          ='999-9999'
    1000000000-9999999999   =' 999) 999-9999' (Prefix '(' );
  Other                      ='Invalid Number';
Data Test; Infile Cards
  Input @1 Phonenum 10.;
Cards;
4144599838
7158452027
303223
6082564128
;
Proc Print Noobs; Format Phonenum Phonefmt.;
```

Produces:

```
Phonenum
(414) 459-9838
(715) 845-2027
Invalid Number
(608) 256-4128
```

Invalue Statement

```
Option Notext82;  
Proc Format;  
    Invalue $Statefmt  
    1='AL'  
    2='AZ';  
Data Test; Infile Cards;  
    Input @1 State : $Statefmt. @3 NetSales 8.;  
Cards;  
1 45678345  
2 22243323  
2 34563323  
;  
Proc Print Noobs;
```

Produces:

State	NetSales
AL	45678345
AZ	22243323
AZ	34563323

The Invalue statement reduces lines of code by associating formats with values during the input statement. See End notes for further information on the Invalue statement.

Table Lookups

Efficient Programming/Recoding

```
Proc Format;  
  Value Zipfmt  
    54405='Abbotsford'  
    54411='Athens'  
    54401='Wausau'  
    53562='Middelton'  
    53081='Shebogyan';  
Data Zip; Infile Cards;  
  Input  @1 ZipCode 5.;  
Cards;  
54405  
54411  
54401  
53562  
53081;  
Data; Set Zip;  
  Length CityName $20.;  
  CityName=Put (ZipCode, Zipfmt.);  
Proc Print;
```

Produces:

ZipCode	CityName
54405	Abbotsford
54411	Athens
54401	Wausau
53562	Middleton
53081	Sheboygan

Table Lookups utilizing Proc Format is an efficient way of reprogramming/recoding data for small datasets (under 10,000 variables). This solution of recoding is one of many (see End Notes for further reference). If the information already exists in a user defined format or format library, it is one of the FASTEST lookup techniques in SAS. Other alternatives would be to use if-then-else statements, SAS merge, or a binary search.

Using Proc Format with Other Procs to Tailor Output

```
Proc Format;
  Value $Statefmt
    'California'='Calif.'
    'Wisconsin'='Wisc.'
    'New Hampshire'='New    Hampsh.';
Data Test; Infile Cards;
  Input @1 Origin  $15. @17 Destin $15. @31 Qty 4.;
Cards;
California      New Hampshire  300
New Hampshire   Wisconsin     100
Wisconsin       California     400
;
Proc Freq;
Table Origin * Destin/NoRow NoCol;
Proc Freq;
Format Destin  $Statefmt.;
Table Origin * Destin/NoRow NoCol;
```

Produces:

Using Proc Format with Other Procs to Tailor Output

How was Proc Format used on the Proc Freq to tailor output? Proc Freq divides label headings up into groups of eight. By creating the Proc Format in groups of eight, the label heading was tailored to a readable output.

Why use Proc Format instead of a label statement? If the format is already created, programming time is saved by using the Proc Format over the label statement.

Other Procs that Proc Format can tailor output for: Calendar, Print, Tabulate, Forms, Gmap, Gplot, Gchart.

Using Proc Format to Avoid Hard-Coding AF Screens

and

Creating Permanent Proc Format Libraries

These topics will be discussed briefly, time permitting, reference to sources for information are provided in the End Notes.

End Notes

Table Lookups

Craig Ray, ***"A Comparison of Table Lookup Techniques"***
SAS Proceedings: SUGI 12 (Cary NC: SAS Institute, Inc., 1987), pp. 40-50.

Craig Ray and James Johnstone, ***"A Survey of Table Lookup Techniques for the SAS System"*** SAS Proceedings: SUGI 14 (Cary NC: SAS Institute, Inc., 1989), pp. 27-33.

Using Proc Format to Tailor other Procs' Output

Hallett German, ***"Demystifying Proc Format"*** SAS Proceedings: SUGI 13
(Cary NC: SAS Institute, Inc., 1988), pp. 1262-1270.

Steve Schulz, ***"Using Formats to Enhance the Appearance of Your Report"*** SAS Proceedings: SUGI 12 (Cary NC: SAS Institute, Inc., 1987), pp. 103-109.

Using Proc Format to Avoid Hard-Coding AF Screens

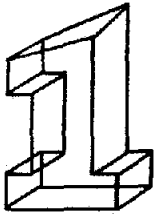
Louise Weiler, ***"Dynamic Screen Generation Using Formats: The Case Against Hard-Coding"*** SAS Proceedings: SUGI 14 (Cary NC: SAS Institute, Inc., 1989), pp. 1516-1520.

Permanent Format Library

SAS Basics Version 5 (Cary NC: SAS Institute, 1985), pp.929-936.

Judith Gabor, ***"Creating Selectively A Permanent SAS Software Format Library from a Homungous Dictionary"*** SAS Proceedings: SUGI 12 (Cary NC: SAS Institute, Inc., 1987), pp.889-892.

Eileen McGivern, ***"Permanent Proc Formats"*** SAS Proceedings: SUGI 12 (Cary NC: SAS Institute, Inc., 1987), pp. 910-915.



**Systems
Seminar
Consultants**

SYSTEMS SEMINAR CONSULTANTS
6014 Gateway Green
Madison, WI 53716
(608)222-7081

October 1, 1990

Gary Plazyk
A.T. Kearney, Inc.
222 S. Riverside Plaza
Chicago, IL 60606

Dear Gary:

Enclosed are the two pages from my Proc Formats presentation which appeared incomplete in the Midwest Users Conference Manual.

I appreciate your interest in my presentation.

As things start gearing up for the planning of the 1991 Midwest Users Conference and you need help, please do not hesitate to call me.

Sincerely,



Kim L. Kolbe

Using Proc Format with Other Procs to Tailor Output

Proc Format;

Value \$Statefmt

'California'='Calif.'

'Wisconsin'='Wisc.'

'New Hampshire'='New Hampsh.';

Data Test; Infile Cards;

Input @1 Origin \$15. @17 Destin \$15. @31 Qty 4.;

Cards;

California New Hampshire 300

New Hampshire Wisconsin 100

Wisconsin California 400

;

Proc Freq;

Table Origin * Destin/NoRow NoCol;

Proc Freq;

Format Destin \$Statefmt.;

Table Origin * Destin/NoRow NoCol;

Produces:

TABLE OF ORIGIN BY DESTIN

ORIGIN	DESTIN			
FREQUENCY PERCENT	CALIFORNIA	NEW HAMPSHIRE	WISCONSIN	TOTAL
CALIFORNIA	0 0.00	1 33.33	0 0.00	33.33 ¹
NEW HAMPSHIRE	0 0.00	0 0.00	1 33.33	33.33 ¹
WISCONSIN	1 33.33	0 0.00	0 0.00	33.33 ¹
TOTAL	1 33.33	1 33.33	1 33.33	100.00 ³

Using Proc Format with Other Procs to Tailor Output

TABLE OF ORIGIN BY DESTIN

ORIGIN	DESTIN			
FREQUENCY PERCENT	CALIF.	NEW HAMPSH.	WISC.	TOTAL
CALIFORNIA	0 0.00	1 33.33	0 0.00	1 33.33
NEW HAMPSHIRE	0 0.00	0 0.00	1 33.33	1 33.33
WISCONSIN	1 33.33	0 0.00	0 0.00	1 33.33
TOTAL	1 33.33	1 33.33	1 33.33	3 100.00

How was Proc Format used on the Proc Freq to tailor output? Proc Freq divides label headings up into groups of eight. By creating the Proc Format in groups of eight, the label heading was tailored to a readable output.

Why use Proc Format instead of a label statement? If the format is already created, programming time is saved by using the Proc Format over the label statement.

Other Procs that Proc Format can tailor output for: Calendar, Print, Tabulate, Forms, Gmap, Gplot, Gchart.