

# PROC FORMAT



## Softco WITH FORMATTED Values

Obs	Name	Division	Years	Sales	Expense	State
1	CHRIS	HARDWARE	Under 5	233.11	94.12	WI
2	MARK	HARDWARE	5-10	298.12	52.65	WI
3	SARAH	SOFTWARE	5-10	301.21	65.17	MN
4	PAT	HARDWARE	Under 5	4009.21	322.12	IL
5	JOHN	HARDWARE	5-10	678.43	150.11	WI
6	WILLIAM	HARDWARE	Over 10	3231.75	644.55	MN
7	ANDREW	SOFTWARE	Over 10	1762.11	476.13	MN
8	BENJAMIN	SOFTWARE	Under 5	201.11	25.21	IL
9	JANET	SOFTWARE	Under 5	98.11	125.32	WI
10	STEVE	HARDWARE	Over 10	6153.32	1507.12	WI
11	JENNIFER	SOFTWARE	Under 5	542.11	134.24	IL
12	JOY	SOFTWARE	Over 10	2442.22	761.98	WI
13	MARY	SOFTWARE	Over 10	5691.78	2452.11	WI
14	TOM	SOFTWARE	5-10	5669.12	798.15	MN
15	BETH	HARDWARE	Over 10	4822.12	982.10	WI



# Welcome to Systems Seminar Consultants, Inc.

---



Systems Seminar Consultants, Inc. is a SAS Alliance Quality Partner™ of SAS. Our team of SAS software experts has a broad base of knowledge and experience working with a variety of complex systems in a number of diverse industry settings. This knowledge and experience is leveraged to help you effectively achieve your business goals.



## Free SAS Newsletter

Our popular publication, *The Missing Semicolon*™, shares SAS software solutions developed by our staff and provides additional technical assistance to our customers.

## SAS Training Services

For over 1,000 students each year, we make SAS software easier to understand, use, and support.

- Public training schedules are posted on our web site.
- Private on-site training options are also available.

# Additional SAS Services

---



## SAS Consulting Services

Our staff of SAS consultants is well-versed in a variety of business areas.

Our specialty areas include:

- Data systems development
- Decision support and business consultation
- Market research and analysis

## SAS Help Desk Services

Our SSC team of experts is available to solve your company's daily SAS problems. Call us to develop a customized support plan that meets your company's needs.

## For More Information

To receive additional information about our services or discuss a specific cost-effective solution for your company:

- Call us at: (608) 278-9964
- Check our Website at: [www.sys-seminar.com](http://www.sys-seminar.com)
- Contact us at: 2997 Yarmouth Greenway Drive, Madison, WI 53711



## Steven J. First, President



- Over 30 years of SAS experience, including hundreds of manufacturing, retail, government, marketing, and financial applications.
- Over 25 years as President and Founder of SSC
- Founder of WISAS and WISUG
- Invited speaker at local, regional, and international user groups

# Objectives

---



Participants completing this session will:

- Understand what SAS formats are.
- Use SAS written formats to display values.
- Be able to build user written formats.
- Use user written formats to make values look better
- Use formats for grouping and aggregating values.
- Create formats from a SAS dataset.
- Create, store and retrieve user written/permanent formats.

# Dataset Used in This Presentation

---



The SOFTCO, INC. company has gathered information about their sales people so that they can create reports and statistics on sales and expenses. It is used to create the SAS dataset Softsale.

CHRIS	H	2	233.11	94.12	WI
MARK	H	5	298.12	52.65	WI
SARAH	S	6	301.21	65.17	MN
PAT	H	4	4009.21	322.12	IL
JOHN	H	7	678.43	150.11	WI
WILLIAM	H	11	3231.75	644.55	MN
ANDREW	S	24	1762.11	476.13	MN
BENJAMIN	S	3	201.11	25.21	IL
JANET	S	1	98.11	125.32	WI
STEVE	H	21	6153.32	1507.12	WI
JENNIFER	S	1	542.11	134.24	IL
JOY	S	12	2442.22	761.98	WI
MARY	S	14	5691.78	2452.11	WI
TOM	S	5	5669.12	798.15	MN
BETH	H	12	4822.12	982.10	WI

# Dataset Used in This Presentation - Input

---



The file contains:

<b>Name</b>	Employee name
<b>Division</b>	Division employed (Hardware or Software)
<b>Years</b>	Years with the company
<b>Sales</b>	Sales for the week
<b>Expense</b>	Expenses for the week
<b>State</b>	Employee location

# PROC PRINT of the Softsale Dataset



## Proc Print of the Softsale Dataset

Obs	Name	Division	Years	Sales	Expense	State
1	CHRIS	H	2	233.11	94.12	WI
2	MARK	H	5	298.12	52.65	WI
3	SARAH	S	6	301.21	65.17	MN
4	PAT	H	4	4009.21	322.12	IL
5	JOHN	H	7	678.43	150.11	WI
6	WILLIAM	H	11	3231.75	644.55	MN
7	ANDREW	S	24	1762.11	476.13	MN
8	BENJAMIN	S	3	201.11	25.21	IL
9	JANET	S	1	98.11	125.32	WI
10	STEVE	H	21	6153.32	1507.12	WI
11	JENNIFER	S	1	542.11	134.24	IL
12	JOY	S	12	2442.22	761.98	WI
13	MARY	S	14	5691.78	2452.11	WI
14	TOM	S	5	5669.12	798.15	MN
15	BETH	H	12	4822.12	982.10	WI

**Notes:** What do the division codes mean? Instead of a string of numbers, could Sales and Expense be displayed with commas and dollar signs?

# Displaying SAS Data Values

---



When we print or display the information in a SAS dataset, by default, SAS determines the best way to display the values. FORMATS allow us to change the way the data values appear and can make them more presentable and easier to understand.

- FORMATS can be specified in the DATA step.
- FORMATS can be overridden in the PROC step.
- You can use a SAS-written format.
- You can create user-written formats.

## Notes:

\*\*\*FORMATS do not change the values in the SAS dataset, they only change the way the values are printed/displayed.\*\*\*

# Displaying SAS Data Values

---



Using formats is a 4-step process.

1. Create or identify the SAS dataset you want to use.
2. Identify the variables that could benefit from formatting.
3. Identify the name of the SAS format you want use or create your own.
4. Associate the variable with the format you want to use for displaying the value.

**Syntax:**     **FORMAT** *variables format . . . ;*

# What is a FORMAT?

---



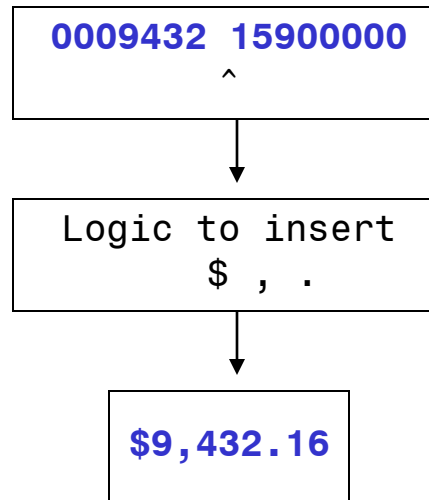
A FORMAT is a routine that a variable value passes through when output.

## Example:

1. Sales variable has a value of 9432.159
2. Print Sales with dollar signs and commas.
3. Allow a total of 10 digits (including \$ , .).
4. Print two digits after the decimal point.

**Format Sales DOLLAR10.2;**

Dollar format



# SAS Provided Formats (Partial list)

---



There are many ways to print/display data values.

## Numeric Formats:

<i>w.</i>	standard numeric
<i>w.d</i>	numeric with decimal
<b>BEST</b> <i>w.</i>	SAS chooses best notation
<b>COMMA</b> <i>w.d</i>	commas in numbers
<b>DOLLAR</b> <i>w.d</i>	insert dollar sign, commas
<b>E</b> <i>w.</i>	scientific notation
<b>FRACT</b> <i>w.</i>	fractions
<b>HEX</b> <i>w.</i>	numeric hexadecimal
<b>IB</b> <i>w.d</i>	integer binary
<b>PD</b> <i>w.d</i>	packed decimal
<b>ROMAN</b> <i>w.</i>	Roman numerals
<b>SSN</b> <i>w.</i>	Social Security Numbers
<b>WORDS</b> <i>w.</i>	numbers as words
<b>Z</b> <i>w.d</i>	print leading zeros
<b>ZD</b> <i>w.d</i>	zoned decimal



## Character Formats:

<b>\$</b> <i>w.</i>	standard character
<b>\$CHAR</b> <i>w.</i>	character with leading blanks
<b>\$HEX</b> <i>w.</i>	character hexadecimal
<b>\$VARYING</b> <i>w.</i>	variable length character

## **Notes:**

- *w.* values specify the width to allow for output.
- For numerics, if you don't give a format, SAS uses BEST*w.* format.
- If you don't specify a format, SAS chooses one.

# An Example Using SAS Provided FORMATS



Formats can be coded in the DATA step.

```
Data Softsale;          fileref
                        Rawin
  Infile Rawin;
  Input  @1  Name      $10.
         @12 Division $1.
         @15 Years    2.
         @19 Sales     7.2
         @28 Expense   7.2
         @36 State     $2.;
```

CHRIS	H	2	233.11	94.12	WI
MARK	H	5	298.12	52.65	WI
SARAH	S	6	301.21	65.17	MN
...					
BETH	H	12	4822.12	982.10	WI

```
FORMAT Name $4. Sales COMMA9.2 Expense COMMA9.2;
```

```
Run;
```

```
Proc Print Data=Softsale;
  Title 'Proc Print WITH FORMATS';
```

```
Run;
```

```
Proc Format
```

# The Resulting Output



Proc Print WITH FORMATS

Obs	Name	Division	Years	Sales	Expense	State
1	CHRI	H	2	233.11	94.12	WI
2	MARK	H	5	298.12	52.65	WI
3	SARA	S	6	301.21	65.17	MN
4	PAT	H	4	4,009.21	322.12	IL
5	JOHN	H	7	678.43	150.11	WI
6	WILL	H	11	3,231.75	644.55	MN
7	ANDR	S	24	1,762.11	476.13	MN
8	BENJ	S	3	201.11	25.21	IL
9	JANE	S	1	98.11	125.32	WI
10	STEV	H	21	6,153.32	1,507.12	WI
11	JENN	S	1	542.11	134.24	IL
12	JOY	S	12	2,442.22	761.98	WI
13	MARY	S	14	5,691.78	2,452.11	WI
14	TOM	S	5	5,669.12	798.15	MN
15	BETH	H	12	4,822.12	982.10	WI

## Notes:

- These formats are stored in the data descriptor.
- All later steps will use these formats unless overridden.

# Another FORMAT Example



Formats can also be coded in the PROC step.

```
Data Softsale;          Fileref
      Infile Rawin;      Rawin
      Input  @1  Name    $10.
              @12 Division $1.
              @15 Years   2.
              @19 Sales    7.2
              @28 Expense  7.2
              @36 State    $2.;
```

CHRIS	H	2	233.11	94.12	WI
MARK	H	5	298.12	52.65	WI
SARAH	S	6	301.21	65.17	MN
...					
BETH	H	12	4822.12	982.10	WI

```
Run;
```

```
Proc Print Data=Softsale;
      Title 'Proc Print WITH FORMATS';
      FORMAT Name $4. Sales COMMA9.2 Expense COMMA9.2;
```

```
Run;
```

# The Resulting Output



Proc Print WITH FORMATS

Obs	Name	Division	Years	Sales	Expense	State
1	CHRI	H	2	233.11	94.12	WI
2	MARK	H	5	298.12	52.65	WI
3	SARA	S	6	301.21	65.17	MN
4	PAT	H	4	4,009.21	322.12	IL
5	JOHN	H	7	678.43	150.11	WI
6	WILL	H	11	3,231.75	644.55	MN
7	ANDR	S	24	1,762.11	476.13	MN
8	BENJ	S	3	201.11	25.21	IL
9	JANE	S	1	98.11	125.32	WI
10	STEV	H	21	6,153.32	1,507.12	WI
11	JENN	S	1	542.11	134.24	IL
12	JOY	S	12	2,442.22	761.98	WI
13	MARY	S	14	5,691.78	2,452.11	WI
14	TOM	S	5	5,669.12	798.15	MN
15	BETH	H	12	4,822.12	982.10	WI

## Notes:

- These FORMATS are used in this step only.

# Creating Your Own FORMATS

---



PROC FORMAT creates FORMATS using Picture and Value statements.

**Syntax:**

## PROC FORMAT options;

**VALUE**      *Name(options)*  
                 *Range1 = 'Label1'*  
                 *Range2 = 'Label2'*  
                 *... ;*

**PICTURE**    *Name(options)*  
                 *Range1 = 'Picture1'(options)*  
                 *Range2 = 'Picture2'(options)*  
                 *... ;*

**RUN:**

**Notes:**

- Keywords: LOW, HIGH, OTHER

# Creating Your Own FORMATS (continued)

---



## Specifying FORMAT names:

- are 1-32 characters long (8 characters in V8 and earlier)
- must start with a letter or underscore
- must begin with \$ for character variables
- cannot end with a number
- do not end in period when created
- DO end in period when used

# The VALUE Statement

---



Print labels corresponding to a range instead of an actual value.

**Ranges can be a single value:**

```
VALUE RESPFMT
  1 = 'Agree'
  2 = 'Disagree';
```

**Ranges can specify a range of values:**

```
VALUE GRADEFMT
  LOW-69 = 'FAILING'
  70-HIGH = 'PASSING';
```

**Ranges can specify a string of values:**

```
VALUE QLTYFMT
  1,10-20,30-HIGH = 'ACCEPT'
  OTHER           = 'REJECT';
```

# The VALUE Statement (continued)

---



Ranges can specify a combination:

```
VALUE GRADEFMT
  1 - 69      = 'FAILING'
  70 - HIGH   = 'PASSING'
  0           = 'INCOMPLETE';
```

Character fields require a "\$" format:

```
VALUE $STATFMT
  'AZ' = 'ARIZONA'
  'TX' = 'TEXAS';
```

```
VALUE $DIVFMT
  'H'   = 'HARDWARE'
  'S'   = 'SOFTWARE'
  OTHER = 'INVALID';
```

# A PROC FORMAT Example

---



Create some formats for SOFTCO.

    Fileref  
    Rawin

CHRIS	H	2	233.11	94.12	WI
MARK	H	5	298.12	52.65	WI
SARAH	S	6	301.21	65.17	MN
...					
BETH	H	12	4822.12	982.10	WI

Data Softsale;

    Infile Rawin;

    Input @1 Name \$10.

        @12 Division \$1.

        @15 Years 2.

        @19 Sales 7.2

        @28 Expense 7.2

        @36 State \$2.;

Run;

# A PROC FORMAT Example (continued)

---



```
PROC FORMAT;  
  VALUE $DIVFMT  
    'H'    = 'HARDWARE '  
    'S'    = 'SOFTWARE '  
    OTHER  = 'INVALID';  
  VALUE YEARFMT  
    LOW - <5    = 'Under 5 '  
        5 - 10  = '5-10 '  
    10< - HIGH = 'Over 10';
```

```
Run;
```

```
Proc Print Data=Softsale;  
  FORMAT Division $DIVFMT.  Years YEARFMT.;  
  Title 'Softco WITH VALUE FORMATS';
```

```
Run;
```

# The Resulting Output



## Softco WITH VALUE FORMATS

Obs	Name	Division	Years	Sales	Expense	State
1	CHRIS	HARDWARE	Under 5	233.11	94.12	WI
2	MARK	HARDWARE	5-10	298.12	52.65	WI
3	SARAH	SOFTWARE	5-10	301.21	65.17	MN
4	PAT	HARDWARE	Under 5	4009.21	322.12	IL
5	JOHN	HARDWARE	5-10	678.43	150.11	WI
6	WILLIAM	HARDWARE	Over 10	3231.75	644.55	MN
7	ANDREW	SOFTWARE	Over 10	1762.11	476.13	MN
8	BENJAMIN	SOFTWARE	Under 5	201.11	25.21	IL
9	JANET	SOFTWARE	Under 5	98.11	125.32	WI
10	STEVE	HARDWARE	Over 10	6153.32	1507.12	WI
11	JENNIFER	SOFTWARE	Under 5	542.11	134.24	IL
12	JOY	SOFTWARE	Over 10	2442.22	761.98	WI
13	MARY	SOFTWARE	Over 10	5691.78	2452.11	WI
14	TOM	SOFTWARE	5-10	5669.12	798.15	MN
15	BETH	HARDWARE	Over 10	4822.12	982.10	WI

## Notes:

- FORMATS do not change the values, they only change the way the values are printed/displayed.

# A PROC FREQ Job Without Formatting



The TABLE YEARS statement produces a count summary based on the actual value of the Years variable.

```
Proc Freq Data=Softsale;  
  Table Years;  
Run;
```

Years	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	2	13.33	2	13.33
2	1	6.67	3	20.00
3	1	6.67	4	26.67
4	1	6.67	5	33.33
5	2	13.33	7	46.67
6	1	6.67	8	53.33
7	1	6.67	9	60.00
11	1	6.67	10	66.67
12	2	13.33	12	80.00
14	1	6.67	13	86.67
21	1	6.67	14	93.33
24	1	6.67	15	100.00

# PROC FREQ With Formatting



A FORMAT causes PROC FREQ to produce a count summary based on the group/formatted value of the Year variable.

## PROC FORMAT;

```
Value YEARFMT LOW - <5    = 'Under 5'  
                5 - 10    = '5-10'  
                10< - HIGH = 'Over 10';
```

Run;

```
Proc Freq Data=Softsale;  
  Table Years;  
  FORMAT Years YEARFMT.;
```

Run;

Years	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Under 5	5	33.33	5	33.33
5-10	4	26.67	9	60.00
Over 10	6	40.00	15	100.00

# Another PROC FREQ With Formatting



```
Proc Format;
```

```
VALUE AMTFMT
```

```
.01-1000 = '.01-1000'
```

```
1000<-2500 = '1001-2500'
```

```
2500<-5000 = '2501-5000'
```

```
5000<-HIGH = 'Greater than 5000';
```

```
Run;
```

```
Proc Freq Data=Softsale;
```

```
Table Sales Expense / List Nocum;
```

```
FORMAT Sales Expense AMTFMT. ;
```

```
Run;
```

Sales	Frequency	Percent
.01-1000	7	46.67
1001-2500	2	13.33
2501-5000	3	20.00
Greater than 5000	3	20.00

Expense	Frequency	Percent
.01-1000	13	86.67
1001-2500	2	13.33

# Formats From a SAS Dataset

---



Hard coding recoded values can be inconvenient if:

- There are many lines to type.
- They may already exist in a file.
- They change often.

Is there some way to read the information needed to create the format from a file rather than typing the values?

# The CNTLIN Option

---



PROC FORMAT can read the values from a SAS dataset as long as the variables in the dataset have special variable names.

## Syntax:

```
PROC FORMAT CNTLIN=sasdataset;  
RUN;
```

# The CNTLIN Dataset – Required Variable Names

---



The CNTLIN dataset must contain the following variables:

<b>FMTNAME</b>	Name of the format	Ex. \$DIVFMT
<b>START</b>	Starting value for range	Ex. 'H'
<b>LABEL</b>	Recoded value to use	Ex. 'Hardware'

OPTIONAL variables include:

<b>END</b>	Ending value for range	Ex. 'I'
------------	------------------------	---------

# A CNTLIN Example

---



We can build a format starting with the flat file Dfile.

```
Data Formds;  
  Infile Dfile;  
  Input @1 START $1.  
         @4 LABEL $12.;  
  Retain FMTNAME '$DIVFMT';
```

Run;

```
Proc Print Data=Formds;  
  Title 'Format Dataset';
```

Run;

```
PROC FORMAT CNTLIN=Formds;  
Run;
```

Fileref  
Dfile

<b>H</b>	<b>Hardware</b>
<b>P</b>	<b>Publications</b>
<b>S</b>	<b>Software</b>

# Output Files and Partial Log



Format Dataset				
Obs	START	LABEL	FMTNAME	
1	H	Hardware	\$DIVFMT	
2	P	Publications	\$DIVFMT	
3	S	Software	\$DIVFMT	

```
PROC FORMAT CNTLIN=Formds;
NOTE: Format $DIVFMT has been output.
Run;
NOTE: PROCEDURE FORMAT used:
      real time          0.04 seconds

NOTE: There were 3 observations read from the data set WORK.FORMDS.
```

# Using the Format

---



Read, format, and print the data as usual.

```
Data Softsale;
  Infile Rawin;
  Input  @1  Name      $10.
         @12 Division $1.
         @15 Years     2.
         @19 Sales     7.2
         @28 Expense   7.2
         @36 State     $2.;

Run;

Proc Print Data=Softsale;
  Title 'Print WITH Formats';
  FORMAT Division $DIVFMT. ;
Run;
```

# The Resulting Output



Print WITH Formats

Obs	Name	Division	Years	Sales	Expense	State
1	CHRIS	Hardware	2	233.11	94.12	WI
2	MARK	Hardware	5	298.12	52.65	WI
3	SARAH	Software	6	301.21	65.17	MN
4	PAT	Hardware	4	4009.21	322.12	IL
5	JOHN	Hardware	7	678.43	150.11	WI
6	WILLIAM	Hardware	11	3231.75	644.55	MN
7	ANDREW	Software	24	1762.11	476.13	MN
8	BENJAMIN	Software	3	201.11	25.21	IL
9	JANET	Software	1	98.11	125.32	WI
10	STEVE	Hardware	21	6153.32	1507.12	WI
11	JENNIFER	Software	1	542.11	134.24	IL
12	JOY	Software	12	2442.22	761.98	WI
13	MARY	Software	14	5691.78	2452.11	WI
14	TOM	Software	5	5669.12	798.15	MN
15	BETH	Hardware	12	4822.12	982.10	WI

# The PICTURE Statement

---



The PICTURE statement creates display layouts for **numeric** variables.

- Pictures are for NUMERIC variables only.
- Pictures are limited to 24 or less positions.
- Pictures are specified with 3 types of characters.

## Types of picture characters:

<b>0</b>	print digit; suppress leading zeros
<b>1-9</b>	print digit; print leading zeros
<b>All others</b>	print that character unchanged



# A Dataset Needing Formatting



Phone numbers, balances, and past-due values.

```
Data Phone1st;  
  Infile Rawin;  
  Input Phonenum Balance Pastdue;  
Run;
```

```
Filedef  
Rawin
```

5552226721	33.22	2216.00
6083442121	16.11	10.00
3412552765	-15.6	-5.20

```
Proc Print Data=Phone1st;  
  Title 'Phone List - WITHOUT Formatting';  
Run;
```

## Phone List - WITHOUT Formatting

OBS	Phonenum	Balance	Pastdue
1	5552226721	33.22	2216.00
2	6083442121	16.11	10.00
3	3412552765	-15.60	-5.20

# Format the Values



Picture formats make the values more presentable.

```
Data Phone1st;  
  Infile Rawin;  
  Input Phonenum Balance Pastdue;  
Run;
```

Filedef  
Rawin

5552226721	33.22	2216.00
6083442121	16.11	10.00
3412552765	-15.6	-5.20

```
PROC FORMAT;
```

```
  PICTURE NEGFMT      LOW - <0      = '00,000.99 - '  
                        0      - HIGH = '00,009.99+' ;  
  PICTURE PHONEFMT  OTHER          = '999-999-9999' ;
```

```
Run;
```

```
Proc Print Data=Phone1st;  
  FORMAT Phonenum      PHONEFMT.;  
  FORMAT Balance Pastdue NEGFMT.;  
  Title 'Phone List - WITH Formatting';
```

```
Run;
```

# Compare The Resulting Output

---



## Phone List - WITHOUT Formatting

Obs	Phonenum	Balance	Pastdue
1	5552226721	33.22	2216.0
2	6083442121	16.11	10.0
3	3412552765	-15.60	-5.2

## Phone List - WITH Formatting

Obs	Phonenum	Balance	Pastdue
1	555-222-6721	33.22+	2,216.00+
2	608-344-2121	16.11+	10.00+
3	341-255-2765	15.60-	5.20-

# Format the Values

---



What if I would like to place the area code of our **10-digit** phone number in parenthesis and precede the dollar values with asterisks.

Phone List with FILL and PREFIX

OBS	Phonenum	Balance	Pastdue
1	(555) 222-6721	***\$33.22+	\$2,216.00+
2	(608) 344-2121	***\$16.11+	***\$10.00+
3	(341) 255-2765	***\$15.60-	****\$5.20-

# FILL and PREFIX Options

---



Fill a field, or place 1 or 2 characters before digits.

## Syntax:

**FILL** = *'character'*

**PREFIX** = *'characters'*

## PROC FORMAT;

```
PICTURE NEGFMT    LOW-<0    = '00,000.99-'
                  ( FILL='*' PREFIX='$' )
                  0-HIGH    = '00,000.99+'
                  ( FILL='*' PREFIX='$' );
```

```
PICTURE PHONEFMT  OTHER    = '0999) 999-9999'
                  ( PREFIX='(' );
```

Run;

# FILL and PREFIX Options Example



```
Proc Print Data=Phone1st;  
  Format Phonenum          PHONEFMT.  
        Balance Pastdue NEGFMT.;  
  Title 'Phone List: Formats Using FILL and PREFIX';  
Run;
```

## Phone List: Formats Using FILL and PREFIX

OBS	Phonenum	Balance	Pastdue
1	(555) 222-6721	***\$33.22+	\$2,216.00+
2	(608) 344-2121	***\$16.11+	***\$10.00+
3	(341) 255-2765	***\$15.60-	****\$5.20-

# Another User Defined FORMAT Example

---



Recode the RESPONSE and display PHONE in an easier/more familiar format.

Fileref	1 6085552424
Rawin	2 3123432424

```
PROC FORMAT;  
  VALUE RESPFMT 1= 'Agree' 2= 'Disagree';  
  PICTURE PHONEFMT LOW-HIGH= '999/999-9999';  
RUN;
```

```
Data Phone1st;  
  Infile Rawin;  
  Input Response Phone;  
Run;
```

```
Proc Print Data=Phone1st;  
  Format Response RESPFMT. Phone PHONEFMT. ;  
  Title 'Proc Print WITH USER FORMATS';
```

```
Run;  
Proc Format
```

# The Resulting Output

---



```
Proc Print WITH USER FORMATS
```

Obs	Response	Phone
1	Agree	608/555-2424
2	Disagree	312/343-2424

## Notes:

- FORMATS do not change the values, they only change the way the values are printed/displayed.

# Creating a Permanent Format

---



To create a format in one job and use it in another, create permanent formats. Simply store the format in a SAS catalog using the Library= option.

```
Libname SSCFLIB 'C:\Temp\';
```

```
PROC FORMAT LIBRARY = SSCFLIB;
```

```
    PICTURE NEGFMT      LOW-<0 = '00,000.99- '  
                                0-HIGH = '00,009.99+' ;
```

```
    PICTURE PHONEFMT    OTHER  = '999-999-9999' ;
```

```
    VALUE    RESPFMT    1= 'Agree'  2= 'Disagree' ;
```

```
Run;
```

# Creating a Permanent Format (Continued)

---



Create the permanent format.

## SAS LOG:

```
55 Libname SSCFLIB 'C:\Temp\';
```

```
NOTE: Libref SSCFLIB was successfully assigned as follows:
```

```
Engine:          V9
```

```
Physical Name:  C:\Temp
```

```
56
```

```
57 PROC FORMAT LIBRARY = SSCFLIB;
```

```
58 PICTURE NEGFMT LOW-<0 = '00,000.99-'
```

```
59 0-HIGH = '00,009.99+' ;
```

```
NOTE: Format NEGFMT has been written to SSCFLIB.FORMATS.
```

```
60 PICTURE PHONEFMT OTHER = '999-999-9999';
```

```
NOTE: Format PHONEFMT has been written to SSCFLIB.FORMATS.
```

```
61 VALUE RESPFMT 1= 'Agree' 2= 'Disagree';
```

```
NOTE: Format RESPFMT has been written to SSCFLIB.FORMATS.
```

```
62 Run;
```

# Using a Permanent Format

---



To use the format, specify the catalog where the format was created, use an OPTIONS statement with the FMTSEARCH= option.

```
Libname SSCFLIB 'C:\Temp\';
```

```
OPTIONS FMTSEARCH=(SSCFLIB);
```

```
Proc Print Data=Phone1st;
```

```
    Format Phonenum          PHONEFMT.;
```

```
    Format Balance Pastdue NEGFMT.;
```

```
    Title 'Phone List - Formatted Using Permanent Formats';
```

```
Run;
```

**Note:** Remember that without the FMTSEARCH option, SAS searches for a format by looking in WORK.FORMATS first, then LIBRARY.FORMATS. With the option, the catalog/s specified in the FMTSEARCH= list are searched first in the order listed, then WORK.FORMATS and then LIBRARY.FORMATS.

# The Resulting Output

---



## Phone List - Formatted Using Permanent Formats

OBS	Phonenum	Balance	Pastdue
1	555-222-6721	33.22+	2,216.00+
2	608-344-2121	16.11+	10.00+
3	341-255-2765	15.60-	5.20-



## **SYSTEMS SEMINAR CONSULTANTS, INC.**

SAS® Training, Consulting, & Help Desk Services

2997 Yarmouth Greenway Drive • Madison, WI 53711

(608) 278-9964 • Fax (608) 278-0065

[www.sys-seminar.com](http://www.sys-seminar.com)



---

Steven First

President

[sfirst@sys-seminar.com](mailto:sfirst@sys-seminar.com)

