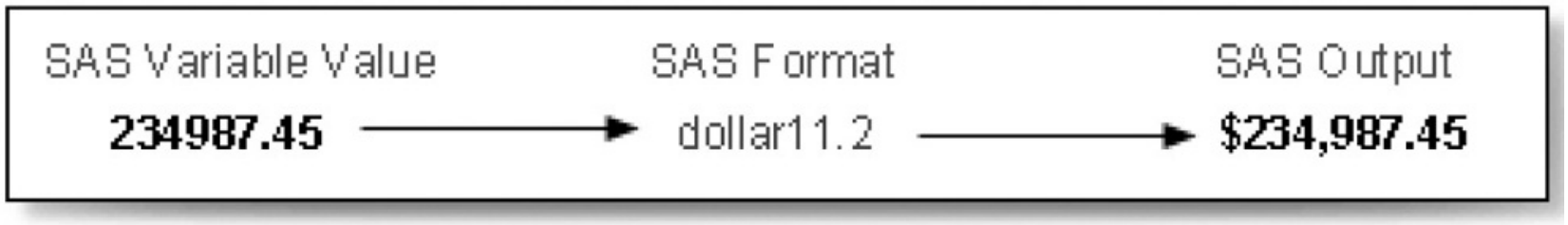
Chapter 7

Creating and Applying User-Defined Formats

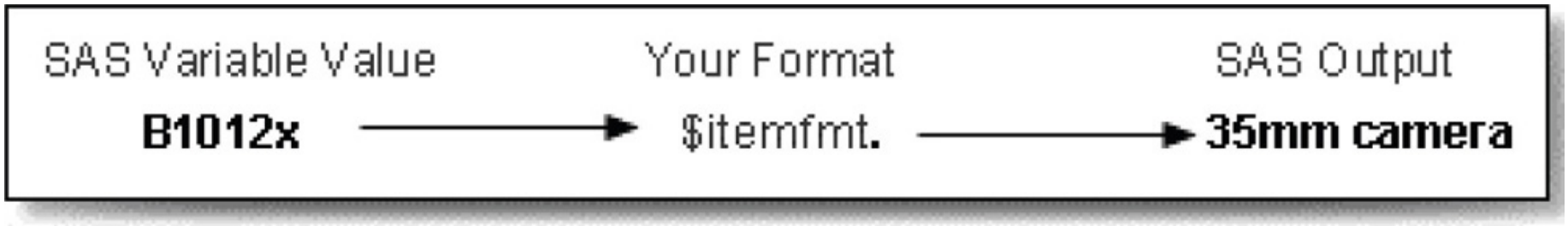
Review

In Creating List Reports, you learned to associate formats with variables, for example:



Create Your Own Formats

You can create custom formats for displaying variable values. For example, you can format a product number so that it is displayed as descriptive text, as shown below.



Topics

- Create your own formats for displaying variable values.
- Permanently store the formats that you create.
- Associate your formats with variables.

Example: Display Coded Variable Values

FirstName	LastName	JobTitle	Salary
Donny	Evans	112	29996.63
Lisa	Helms	105	18567.23
John	Higgins	111	25309.24
Amy	Larson	113	32696.78
Mary	Moore	112	28945.89
Jason	Powell	103	35099.55

The values for JobTitle are coded, and they are not easily interpreted.

FirstName	LastName	JobTitle	Salary
Donny	Evans	technical writer	23936.63
Lisa	Helms	text processor	18567.23
John	Higgins	assoc. technical writer	25303.24
Amy	Larson	senior technical writer	32636.78
Mary	Moore	technical writer	28945.89
Jason	Powell	manager	35033.55

More descriptive values are displayed using a user defined format.

The PROC FORMAT Procedure

You can use the FORMAT procedure to define your own formats for displaying values of variables.

Syntax:

PROC FORMAT *<options>;*

where *options* include

- **LIBRARY**=*libref* specifies the libref for a library to contain a permanent catalog of user-defined formats. If you do not specify the *LIBRARY*= option, the formats are stored in a default format catalog named Work.Formats.
- **FMTLIB** displays a list of all of the formats in your catalog, along with descriptions of their values.

Permanently Store Your Formats

```
libname library 'c:\sas\formats\lib';
proc format library=library;
/* you can also write:
    proc format lib=library;  */
… ;
run;
```

Any format that you create in this PROC FORMAT step is stored in a permanent format catalog called Library.Formats

Also, you can specify a catalog name in the LIBRARY= option, and you can store formats in any catalog. The catalog name must conform to SAS naming conventions.

proc format lib=library.catalog;

Permanently Store Your Formats

If you want to store your format in a library rather than libref **library**, you should add an options statement before proc format procedure.

Example:

```
LIBNAME      project 'c:\sas\chapter7';
OPTIONS      fmtsearch=(project);
/* fmtsearch= (catalog-specifications)  */
PROC FORMAT          library=project;
…

Run;
```

Define a Unique Format

Overview

You use the **VALUE** statement to define a format for displaying one or more values.

VALUE *format-name*
range1='label1'
range2='label2'
…;

where *format-name*

- must begin with a dollar sign ($) if the format applies to character data
- must be a valid SAS name (up to 32 characters, including $)
- 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

range: specifies one or more variable values

label: a text in quotation marks that will be displayed instead of the original variable value

Define a Unique Format

The VALUE range can specify:

- A single value: e.g., 25 or 'M'
- A range of numeric values: e.g., 1-50
- A range of character values: e.g., 'A'-'G'
- A list of unique values separated by commas: 1, 5, 7 or 'A', 'C', 'L'

Define a Unique Format

When specifying a label for displaying each range, remember to

- enclose the label in quotation marks
- limit the label to 256 characters
- use two single quotation marks if you want an apostrophe to appear in the label when enclosed with single quotes, or use double quotes to enclose the label containing an apostrophe:

```
00='employee''s jobtitle unknown';
00="employee's jobtitle unknown";
```

Define a Unique Format

Example: specifying character or numeric values

```
Proc format library=library;
        Value $gradef
                'A'='Good'
                'B'-'D'='Fair'
                'F'='Poor'
                'I','U'='See Instructor';
run;
```

```
Proc format library=library;
        Value genderf
                0='Male'
                1='Female';
run;
```

Define a Unique Format

Example: specifying a range of values

```
proc format library=library;
    value agefmt
          0-<13='Child'
          13-<20='Teenager'
          20-<65='Adult'
          65-100='Senior';
Run;
```

```
proc format library=library;
    value agefmt
          LOW-<13='Child'
          13-<20='Teenager'
          20-<65='Adult'
          65-HIGH='Senior'
          OTHER='unknown';
run;
```

Note : using **<13='Child'** will include missing values.

LOW and HIGH keywords for lower and upper limits of a variable's value range; the keyword LOW does not include missing numeric values. The keyword OTHER is for missing values as well as any values that are not specifically addressed in a range.

Define Multiple Formats

```
proc format library=library;
        value genderf
                0='Male'
                1='Female';
        value agefmt
                LOW-<13 ='Child'
                13-<20='Teenager'
                20-<65='Adult'
                65-HIGH='Senior'
                OTHER='unknown';
run;
```

Associating User-Defined Formats to Variables

```
libname library 'c:\sas\formats\lib';
proc format lib=library;
    value jobfmt
        103='manager'
        105='text processor'
        111='assoc. technical writer'
        112='technical writer'
        113='senior technical writers';
run;
libname perm 'c:\data\perm';
filename empdata 'c:\data\temp\newhires.txt';
data perm.employee;
    infile empdata;
    input @9 FirstName $5. @1 LastName $7. +7
          JobTitle 3. @19 Salary comma9.;
    format salary comma9.2 jobtitle jobfmt.;
run;
```

SAS searches for the format **jobfmt** in two libraries, in this order:

- the temporary library referenced by the libref **Work**
- a permanent library referenced by the libref **Library**.

Assign the user-defined format, which **invariably requires a period** at the end of the format name

Using the User-Defined Format: the Result

FirstName	LastName	JobTitle	Salary
Donny	Evans	112	29996.63
Lisa	Helms	105	18567.23
John	Higgins	111	25309.24
Amy	Larson	113	32696.78
Mary	Moore	112	28945.89
Jason	Powell	103	35099.55

FirstName	LastName	JobTitle	Salary
Donny	Evans	technical writer	23,936.63
Lisa	Helms	text processor	18,567.23
John	Higgins	assoc. technical writer	25,303.24
Amy	Larson	senior technical writer	32,636.78
Mary	Moore	technical writer	28,945.89
Jason	Powell	manager	35,033.55

Displaying a List of All the Formats in a Catalog

```
libname library 'c:\sas\formats\lib';
proc format library=library fmtlib;
run;
```

SAS Output

Format Name: JobFmt Length: 23 Number of Values: 5
Min Length: 1 Max Length: 40 Default Length: 23 Fuzz: Std

START	END	LABEL (VER. 9.00 29AUG2002:11:13:14)
103	103	manager
105	105	text processor
111	111	assoc. technical writer
112	112	technical writer
113	113	senior technical writer

Deleting Formats

There are two ways to delete a format

- Manually delete from the EXPLORER window in SAS
- Use the PROC CATALOG procedure to delete a format

```
proc catalog catalog=library.formats;
        delete agefmt.format;
run;
```