**Chapter VII – Creating and Applying User-Defined Formats**

**PROC FOEMAT** step(Custom Format)

Using **FORMAT** procedure, you can define your own formats for variable, temporarily or permanently

注意：FORMAT statement is the format that already programed by the SAS system

Basic code:

**PROC FORMAT** <*options*>**;**

where *options* include

**LIBRARY**=*libref* specifies the libref for a SAS data library to contain a permanent catalog of user-defined formats, without LIBRARY statement, this format is going to be temporary (Quiz 1)

**FMTLIB** displays a list of all of the formats in your catalog, along with descriptions of their values. (Quiz 10)

* The format options are stored in a format catalog, if data library dose not contain a format catalog, SAS will create one automatically
* If you do not specify the **LIBRARY**= option, the formats are stored in a default format catalog named Work.Formats as a temporarily format

1. Permanently Storing Your Formats

You can store your formats in a permanent format catalog named Formats when you specify the LIBRARY= option in the PROC FORMAT statement.

Eg:

**PROC FORMAT LIBRARY**=libref;

* First, need a **LIBNAME** statement that associates the libref with the permanent SAS data library in which the format catalog is to be stored.
* Than using **PROC FORMAT s**tep to store the permanent format catalog

Eg:

**LIBNAME** **LIBRARY** 'c:\sas\formats\lib';

**PROC FORMAT LIBRARY** =library;

... ;

**RUN**;

1. Defining a Unique Format

**VALUE** statement

Use to define a format for displaying one or more values, and multiple **VALUE** statement can be written in a same **PROC FORMAT** step

Basic code:

**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

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 (Quiz 3)

Eg:

**PROC FORMAT LIB**=library;

**VALUE** jobfmt

103='manager'

105='text processor'

111='assoc. technical writer'

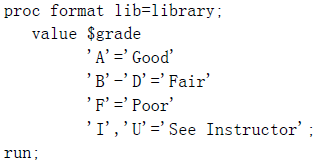
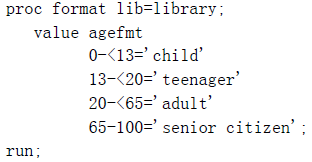
112='technical writer'

113='senior technical writer';

**RUN**;

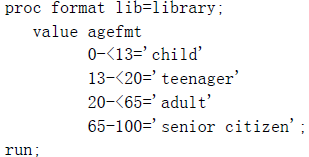
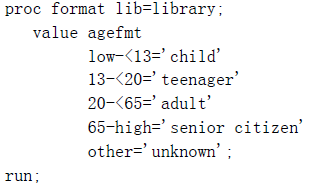
* When the specified values are character values, they must be enclosed in quotation marks and must match the case of the variable's values. The format's name must also start with a dollar sign ($). When the specified values are numeric values, they are not enclosed in quotation marks, and the format's name should not begin with a dollar sign ($).

Eg:

* Specifying Value Range

Eg:

* The keywords **LOW** and **HIGH** to specify the lower and upper limits of a variable's value range.
* The keyword **OTHER** can be used to label missing values

**Program Syntax:**

**LIBNAME** *libref 'SAS-data-library'***;**

**PROC FORMAT LIBRARY=***libref* **FMTLIB;**

**VALUE** *format-name*

*range1='label1'*

*range2 'label2'*

...**;**

**RUN;**

**DATA** *SAS-data-set;*

**INFILE** *data-file;*

**INPUT** *pointer variable-name informat.;*

**FORMAT** *variable(s) format-name.;*

**RUN;**

**Sample Program:**

**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 writer';

**RUN**;

**LIBNAME** perm 'c:\data\perm';

**FILENAME** empdata 'c:\data\temp\newhires.txt';

**DATA** perm.employee;

**INFILR** empdata;

**INPUT** @9 FirstName $5. @1 LastName $7. +7 JobTitle 3.

@19 Salary comma9.;

**FORMAT** salary comma9.2 jobtitle jobfmt.;

**RUN**;

练习

1. If you don't specify the LIBRARY= option, your formats are stored in Work.Formats, and they exist ...
2. only for the current procedure.
3. only for the current DATA step.
4. only for the current SAS session.
5. permanently.
6. When creating a format with the VALUE statement, the new format's name

* cannot end with a number
* cannot end with a period
* cannot be the name of a SAS format, and...

1. cannot be the name of a data set variable.
2. must be at least two characters long.
3. must be at least eight characters long.
4. must begin with a dollar sign ($) if used with a character variable.
5. How many characters can be used in a label?
6. 40
7. 96
8. 200
9. 256
10. You can place the FORMAT statement in either a DATA step or a PROC step. What happens when you place it in

a DATA step?

1. You temporarily associate the formats with variables.
2. You permanently associate the formats with variables.
3. You replace the original data with the format labels.
4. You make the formats available to other data sets.
5. 9. The format JOBFMT was created in a FORMAT procedure. Which FORMAT statement will apply it to the

variable JobTitle in the program output?

1. format jobtitle jobfmt;
2. format jobtitle jobfmt.;
3. format jobtitle=jobfmt;
4. format jobtitle='jobfmt';