

Converting a Code Book to a SAS Format Library

If you have an on-line version of a code book, with some work, you can often convert the code book to a SAS format library. This can save you a lot of typing if you need value labels for a SAS data set. As an example, consider a small portion from one of the PUMS code books (Note: the first line is to be used as a ruler and is not part of the code book):

123456789012345678901234567890123456789012345678901234567890123456

D WATER 1 63
 Source of water

V		0	.N/A (GQ)
V		1	.Public system or private company
V		2	.Individual drilled well
V		3	.Individual dug well
V		4	.Other source such as a spring,etc.

D SEWAGE 1 64
 Sewage disposal

V		0	.N/A (GQ)
V		1	.Public sewer
V		2	.Septic tank or cesspool
V		3	.Other means

D YRBUILT 1 65
 When structure first built

V		0	.N/A (GQ)
V		1	.1989 or 1990
V		2	.1985 to 1988
V		3	.1980 to 1984
V		4	.1970 to 1979
V		5	.1960 to 1969
V		6	.1950 to 1959
V		7	.1940 to 1949
V		8	.1939 or earlier

In order to convert these lines to a SAS format library, you must be able to create a SAS data set from the code book which contains three variables:

FMTNAME a name for the SAS format.

START a numeric value the variable contains

LABEL the value label the numeric value (START) is to be assigned

Columns 3-10 from lines beginning with D could be read in as the FMTNAME variable, column 29 from lines beginning with V could be read in as the START variable, and columns 32-65 from lines beginning with V could be read in as the LABEL variable.

The conversion process requires three steps and are described in this handout:

- 1) Edit the file containing the code book to ensure that it gets read properly.
- 2) Create a SAS data set from the code book.
- 3) Use the FORMAT procedure to convert the data set to a SAS format library.

Step1: Edit the Code Book

Most code books are going to require some editing before they can be converted to a SAS format library. It's not possible to include everything here that may require editing but below are a few of the more common things:

- SAS format names (FMTNAME) cannot end in a number. Many fields identifying the format name in code books end in a number. It's true that you could use SAS functions to perhaps replace numbers with other characters but you run the risk of creating multiple formats with the same name.
- SAS format names must follow the rules for SAS names - they must begin with a letter or underscore and can be no longer than eight character. Also, special characters like dashes (-) and percent signs (%) are not allowed.
- Labels can be no longer than 40 characters.
- Fields that contain ranges for the START variable like the one below need to be modified or deleted from the code book to be converted:

```

D ELECCOST      4      69
  Electricity (yearly cost)
V                0000 .N/A (GQ/vacant)
V                0001 .Included in rent or condo
fee
V                0002 .No charge or elect. not
used
V                0003..3099 .$3 to $3,099
V                3100 .Topcode
V                3101+ . $3101 or more

```

In the above example, the line containing the range 0003..3099 needs to be deleted.

- Fields that contain a character in the field being read in as START need to be modified or deleted because START is a numeric variable. In the example above, the line containing the value 3101+ needs to be deleted.

Step 2: Create the SAS Data Set

SAS refers to a SAS data set that is to be converted to a SAS format library as a *control data set*. Control data sets must contain the variables FMTNAME, START, and LABEL as described above. The program below is an example of a SAS data step that reads a

code book like the one on the first page and creates a control data set:

```
data one;
  drop check;
  infile "pums.dct" missover pad;
  input check $ @;
  if check="D" then do;
    input fmtname $ 3-10;
    retain fmtname;
  end;
  else if check="V" then do;
    input start 20-29 label $ 32-80;
    end;
  else delete;
  if start eq . then delete;
  if check="V" then output;
run;
```

The data set is displayed below:

FMTNAME	START	LABEL
WATER	0	N/A (GQ)
WATER	1	Public system or private company
WATER	2	Individual drilled well
WATER	3	Individual dug well
WATER	4	Other source such as a spring, etc
SEWAGE	0	N/A (GQ)
SEWAGE	1	Public sewer
SEWAGE	2	Septic tank or cesspool
SEWAGE	3	Other means
YRBUILT	0	N/A (GQ)
YRBUILT	1	1989 or 1990
YRBUILT	2	1985 to 1988
YRBUILT	3	1980 to 1984
YRBUILT	4	1970 to 1979
YRBUILT	5	1960 to 1969
YRBUILT	6	1950 to 1959
YRBUILT	7	1940 to 1949
YRBUILT	8	1939 or earlier

Step 3: Convert the Control Data Set to a SAS Format Library

Use the FORMAT procedure with the CNTLIN= option to convert the control data set to a SAS format library. The following code would convert the control data set created above:

```
proc format cntlin=one fmtlib;
run;
```

The FMTLIB option requests PROC FORMAT to print the contents of the library to the output file. The output is shown below:

FORMAT NAME: SEWAGE LENGTH: 23 NUMBER OF VALUES: 4		
MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 23 FUZZ: STD		
START	END	LABEL (VER. 6.12 02DEC97:14:20:58)
0		N/A (GQ)
1		Public sewer
2		Septic tank or cesspool
3		Other means

FORMAT NAME: WATER LENGTH: 41 NUMBER OF VALUES: 5		
MIN LENGTH: 1 MAX LENGTH: 41 DEFAULT LENGTH 41 FUZZ: STD		
START	END	LABEL (VER. 6.12 02DEC97:14:20:58)
0		N/A (GQ)
1		Public system or private company
2		Individual drilled well
3		Individual dug well
4		Other source such as a spring, creek, et

FORMAT NAME: YRBUILT LENGTH: 15 NUMBER OF VALUES: 9		
MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 15 FUZZ: STD		
START	END	LABEL (VER. 6.12 02DEC97:14:20:58)
0		N/A (GQ)
1		1989 or 1990
2		1985 to 1988
3		1980 to 1984
4		1970 to 1979
5		1960 to 1969
6		1950 to 1959
7		1940 to 1949
8		1939 or earlier

To make the format library permanent, add a LIBNAME statement and modify the call to PROC FORMAT:

```
libname library "~/myformats";
proc format cntlin=one fmtlib library=library;
```

```
run;
```

Then to use the value labels created above, use a FORMAT statement as the following program illustrates:

```
proc print data=read.mypums;
  var water sewage yrbuilt;
  format water water.
         sewage sewage.
         yrbuilt yrbuilt.;
run;
```

A partial listing of the PROC PRINT output is listed below:

WATER	SEWAGE	YRBUILT
Public system or private company	Septic tank or cesspool	1980 to 1984
N/A (GQ)	Other means	1939 or earlier

SOCIAL SCIENCE COMPUTING COOPERATIVE