A Documentation And Retrieval Strategy Using The SAS* System

John Podgurski
Syracuse University

INTRODUCTION

It is possible using the SAS* system to attach extensive documentation information to SAS data sets. This is done using the labeling features available in the SAS data step. Longer labels can be added for the dataset name and variable names when a permanent SAS dataset is created. Descriptive labels for the values of variables can also be created and permanently stored in format libraries.

Starting with Version 5 the SAS system can be used to output SAS datasets containing the labeling information. It is then possible to merge other information to the documentation SAS datasets, and develop customized data dictionaries or other reports.

In addition to customized reports on the documentation, it is possible to use the INDEX function in SAS to search for common character strings within the values of variables in the documentation data sets. This can be the basis for setting up a retrieval system to search single SAS data sets or multiple data sets being used in a research project or management information system. As documentation information becomes more complex, the use of search and retrieval mechanisms built into most data base management systems would be preferable.

An overall strategy for documentation and retrieval will be presented. Suggestions will be given on how to combine elements of the SAS system to implement this strategy for projects of varying complexity.

DOCUMENTATION

Labeling SAS Programs

There are several advantages to preparing a thoroughly labeled permanent SAS dataset-format library combination. These include ease of use, self-descriptive output, less computer processing time, and significant savings in the SAS user's time and effort. One further advantage, which will be emphasized in this paper, is that the documentation information can be made accessible as data for search and retrieval.

SAS programs to create permanent labeled SAS dataset-format library combinations can be developed several ways within the SAS system. The program lines necessary for the INPUT, LABEL, FORMAT, and PROC FORMAT statements can be entered separately for each statement in the SAS data step. PROC FSEDIT with the DATA=NEW option can be used to prepare all but the PROC FORMAT statement. PROC FORMAT program lines can then be entered and processed in a separate SAS data step.

Procedures for establishing the connection between permanent SAS datasets and format libraries have been described elsewhere. Useful information is contained in the various manuals serving as operating system companions to SAS. Detailed instructions on the process were published as part of the Proceedings of SUGI 12. That paper was entitled "A Codebook Template for the Creation and Use of Permanent SAS Datasets and Format Libraries (CMS* Application)". What must be remembered is that two steps are required. The permanent SAS dataset must be created in a job in which temporary formats are created using a PROC FORMAT statement. The permanent format library must be created separately.

If one has an already existing SAS dataset, the ATTRIB statement can be used in a data step to add variable labels or format names to the directory of the SAS dataset. Similarly, PROC FSEDIT can be used to update the directory of a SAS dataset. The two step process to establish the connection between the permanent SAS dataset and permanent SAS format library can be repeated following changes in the variables or their attributes in a SAS dataset.

Documented SAS Datasets

The programming techniques described above are the most commonly used in the creation of documented SAS datasets. Other techniques use the SAS data step to edit systematically arranged information about data and produce SAS program lines, which are then used to create a SAS dataset. The codebook template paper contains an example of such a technique.
Raw data collected by third parties is often accompanied by a formatted data dictionary. It is possible to use SAS programming techniques similar to those used in the codebook template paper to read the lines of a data dictionary and create a SAS program for inputting the raw data. The extent to which labeling information would be included in the resulting SAS dataset would depend on the amount of labeling information contained within the data dictionary or other sources of information about the data.

The CONVERT procedure within SAS can be used to create SAS datasets from system files created by other programming systems. In several cases labeling information is included in the newly created SAS dataset. Where not all labeling information included in the original system file is incorporated into the SAS dataset resulting from the use of PROC CONVERT, it might be best for documentation purposes to edit a data dictionary created by the other system and then create a SAS program from the resulting program lines. This would make more complete use of SAS labeling features possible.

PROC CONTENTS Output SAS Dataset

Starting with Version 5 of SAS an output dataset can be created with PROC CONTENTS. This makes it possible to use the documentation features of SAS in several new ways. The variables from the original SAS dataset become observations in the CONTENTS dataset. The variables in the CONTENTS dataset include the name of the variable, its longer label, and its format. Because it is a SAS dataset, it can be merged with other SAS datasets with similar observations. Researchers using survey instruments can now add the text of questions to a CONTENTS dataset as new variables. Notes descriptive of variables are often included with census tapes. This information can also be added as variables to a CONTENTS dataset. The CONTENTS output dataset will contain 17 variables. For a selected list of key variables see Figure 1.

The descriptive variables in the CONTENTS dataset will contain words subject to search. Variable names can be identified which are associated with the descriptive variables. It will then be possible to return to the original dataset(s) to generate desired statistical information about selected variables.

Figure 1

<table>
<thead>
<tr>
<th>Key Variables: CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMLABEL dataset label</td>
</tr>
<tr>
<td>NAME variable name</td>
</tr>
<tr>
<td>LABEL variable label</td>
</tr>
<tr>
<td>FORMAT variable format</td>
</tr>
</tbody>
</table>

Suggested Additional Variables

<table>
<thead>
<tr>
<th>QUESTEXT text of questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMENTS notes in comments file</td>
</tr>
</tbody>
</table>

PROC FMTLIB Output Dataset

PROC FMTLIB is a procedure described in the SUGI Supplemental Library User's Guide, Version 5 Edition. Starting with Version 5 it is possible to create an output dataset from a permanent SAS format library. The FMTLIB output dataset and CONTENTS output dataset contain a variable in common. This variable is the format name. It is possible using the MERGE statement to include information from the FMTLIB output dataset with the CONTENTS output dataset to create a new SAS dataset. Let's call the new dataset "COMBINED". Each value label is a separate variable in the combined SAS dataset and can be searched for desired character strings. The FMTLIB output dataset will contain 14 variables. For a selected list of key variables see Figure 2.

Another desirable merge of dataset information could be useful. In this merge variables in datasets with common formats are first identified. The variable names and dataset names are then added as variables to the FMTLIB output dataset. They are merged by format name. This merge would be useful because it would identify variables from the same or different datasets, which share the same format. This would be a strong clue to researchers that a closer look at these variables could be fruitful.

Figure 2

<table>
<thead>
<tr>
<th>Key Variables: FMTLIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMTNAME format module name</td>
</tr>
<tr>
<td>LABEL value label</td>
</tr>
<tr>
<td>TYPE type of format</td>
</tr>
</tbody>
</table>

Suggested Additional Variables

<table>
<thead>
<tr>
<th>MEMLOC datasets where format used</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARNAMES variables using format</td>
</tr>
</tbody>
</table>
Data Dictionary Templates

Once a "COMBINED" SAS dataset is available, it is possible to prepare a report on the information about data in any of several data dictionary formats. The report format, or template, could be based on personal preference or as part of data base design considerations.

A fully labeled SAS dataset-format library combination would contain information on a variable including its name, longer label, format name, value labels, variable type, position in the dataset, etc. If the information were to be put into one SAS dataset as in the "COMBINED" SAS dataset outlined above, it would be possible to use the PUT statement to order the report according to one's preference. In such a data dictionary it would usually be desirable to place all of the information about a variable in one location within the report. Other reports on the dataset labels and variables identified with specific formats could also be produced.

An AF menu could be prepared to allow users to select the type of report or subset of a report they prefer. A model of the SAS programming techniques appropriate for preparing such a report is presented in the codebook template paper.

RETRIEVAL

Use of Data Base Management Systems

The scope of the retrieval challenge determines the tools used for retrieval. Large scale retrieval projects require a full fledged data base management system, such as System 2000®. It is conceivable that there could be hundreds of CONTENTS and/or FMTLIB datasets, which it would be desirable to search for variables appropriate to a given research question. It would then be necessary to retrieve the appropriate information about variables in the datasets and store it. Software tools for such operations are built into a number of data base management systems.

Base SAS Data Management

Tools for search and retrieval tasks of a less complicated nature are readily available using the data step in Base SAS software. Once a CONTENTS SAS dataset has been created, variables containing descriptive information can be searched for the existence of selected character strings. The following example should serve to make the process clear. The data set chosen for the example is the National Opinion Research Council (NORC) dataset. This dataset contains survey data on 1157 variables. The data was collected between the years 1972 and 1987. A SAS dataset was created which included variable labels.

It was decided to retrieve information on variables which had the character string 'race' in their label. First it was necessary to prepare a CONTENTS output dataset. Then, using a SET statement with a SUBSETTING IF statement, followed by a simple PROC PRINT, it was possible to prepare a report on the desired subset. The INDEX function in SAS was used in the SUBSETTING IF statement to extract only those observations which had the character string 'race' in the variable label. A SAS log and SAS Listing file for this example program are presented in Appendix I. The program for this simple retrieval example is presented below in Figure 3. Using the AP menu facility in combination with the SAS macro language it would be possible to allow users to select data sets from a library, search strings, prepare simple reports and undertake statistical analyses of variables found in a search. As has been mentioned earlier, large scale application retrieval would best be done using the tools in a full fledged data base management system.

Figure 3 Simple Retrieval Program

PROC CONTENTS DATA=IN.NORC87 NOPRINT OUT=SASCNT.NORC87;
DATA RETRIEVE;
SET SASCNT.NORC87 (KEEP=NAME LABEL);
IF INDEX(LABEL,'RACE');
PROC PRINT;

CONCLUSIONS

Strategy

This documentation and retrieval strategy is made possible, because of the wide variety of programming tools available within the SAS system. The strategy is a synthesis of ideas for documentation and retrieval which can be implemented due to the variety and power of these programming tools. Tools identified have included the labeling features, merge capabilities, the index function, attribute handling, and output file features of the SAS data step. Central to this strategy is the output dataset feature for PROC CONTENTS and PROC FMTLIB. The SAS/PSP® and SAS/AF®
products can be useful in developing user friendly applications of this strategy. Finally, System 2000 software can be used to implement this strategy for complex projects.

Model

A model of this strategy is presented in Appendix II. It develops a picture of how the elements of the strategy, as discussed in this paper, can be seen to fit together. Refinement of the strategy and elaboration of the model await further experimentation with practical applications.

References


Acknowledgements

This paper would not have been possible without the support and encouragement of Ron Kalinoski and Academic Computing Services at Syracuse University.

For more information contact the author at the following address:

Academic Computing Services
732 Ostrom Avenue
Syracuse University
Syracuse, New York 13244

SAS*, System 2000*, SAS/FSP* and SAS/AF* are registered trademarks or trademarks of SAS Institute, Inc.

VM/CMS* is a registered trademark of International Business Machines Corporation.
APPENDIX I  SIMPLE RETRIEVAL EXAMPLE

1  /***************************************************************
2  THIS PROGRAM WILL ACCESS A SAS DATA LIBRARY RESIDENT ON MVS DISK,
3  CREATE A SAS CONTENTS DATA SET ON CMS, SEARCH A VARIABLE IN THAT
4  DATASET FOR A SPECIFIED CHARACTER STRING, RETRIEVE A SUBSET OF
5  THE CONTENTS DATASET BASED ON THE OCCURRENCE OF THE STRING, AND
6  THEN PRINT A REPORT.
7  ***************************************************************/

8  OPTIONS LINESIZE = 78 ;
9  CMS OSVOL MVS804 F ;
10  CMS FILEDEF IN DISK DUMMY DUMMY F DSN SOC SAS NORC7287 ;
11  PROC CONTENTS DATA=IN.NORC87 NOPRINT OUT=SASCNT.NORC87 ;
12  NOTE. THE DATA SET SASCNT.NORC87 HAS 1157 OBSERVATIONS AND 18 VARIABLES.
13  NOTE. THE PROCEDURE CONTENTS USED 0.64 SECONDS AND 672K.
14  DATA RETRIEVE ;
15  SET SASCNT.NORC87 (KEEP = NAME LABEL ) ;
16  IF INDEX(LABEL,'RACE') ;
17  TITLE 1 'SEARCH OF LABEL FOR "RACE"' ;
18  TITLE2 'LISTING FILE' ;
19  NOTE. DATA SET WORK.RETRIEVE HAS 17 OBSERVATIONS AND 2 VARIABLES.
20  NOTE. THE DATA STATEMENT USED 0.08 SECONDS AND 224K.
21  PROC PRINT;
22  NOTE. THE PROCEDURE PRINT USED 0.05 SECONDS AND 224K
23  AND PRINTED PAGE 1.
24  NOTE: SAS USED 672K MEMORY.

SEARCH OF LABEL FOR 'RACE'
LISTING FILE

<table>
<thead>
<tr>
<th>OBS</th>
<th>NAME</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MORENUKE</td>
<td>POSSIBILITY OF INCREASING ARMS RACE</td>
</tr>
<tr>
<td>2</td>
<td>RACCHURH</td>
<td>ATTEND CHURCH WITH OTHER RACE</td>
</tr>
<tr>
<td>3</td>
<td>RACLOS</td>
<td>ANY OPP. RACE LIVING CLOSE TO R</td>
</tr>
<tr>
<td>4</td>
<td>RACDIN</td>
<td>OBJECT TO OPP. RACE HOME FOR DINNER</td>
</tr>
<tr>
<td>5</td>
<td>RADIS</td>
<td>HOW FAR ARE CLOSEST OPP. RACE</td>
</tr>
<tr>
<td>6</td>
<td>RACE</td>
<td>RACE OF RESPONDENT</td>
</tr>
<tr>
<td>7</td>
<td>RACE1</td>
<td>RACE OF PERSON NUMBER 1</td>
</tr>
<tr>
<td>8</td>
<td>RACPEW</td>
<td>CHILDREN TO SCHOOL WITH OPPOSITE RACE</td>
</tr>
<tr>
<td>9</td>
<td>RACHAP</td>
<td>CHILDREN TO SCHOOL HALF OPPOSITE RACE</td>
</tr>
<tr>
<td>10</td>
<td>RACHOME</td>
<td>OPP. RACE HOME FOR DINNER RECENTLY</td>
</tr>
<tr>
<td>11</td>
<td>RACIMP</td>
<td>IMPORTANCE OF RACE relations issues to R</td>
</tr>
<tr>
<td>12</td>
<td>RACINFO</td>
<td>HOW MUCH INFO DOES R HAVE ON RACE ISSUES</td>
</tr>
<tr>
<td>13</td>
<td>RACLIVE</td>
<td>ANY OPP. RACE IN NEIGHBORHOOD</td>
</tr>
<tr>
<td>14</td>
<td>RACMAREL</td>
<td>WHAT IF RELATIVE MARRIED OTHER RACE</td>
</tr>
<tr>
<td>15</td>
<td>RACMOST</td>
<td>CHILDREN TO SCHOOL MOSTLY OPPOSITE RACE</td>
</tr>
<tr>
<td>16</td>
<td>RACTHINK</td>
<td>HOW OPT R &amp; FRNDS THINK ABT RACE ISSUES</td>
</tr>
<tr>
<td>17</td>
<td>RACWRITE</td>
<td>R EVER WRITTEN OFFICIAL ABT RACE ISSUES</td>
</tr>
</tbody>
</table>

1394