A META-DATABASE IMPLEMENTATION STRATEGY FOR DOCUMENTATION AND RETRIEVAL USING THE SAS* SYSTEM

John Podgurski, University of Rochester

Introduction

Data about data (meta-data) is the subject of this paper. The lack of a means for creating meta-data and being efficient in its retrieval is seen as a problem. This paper is an attempt to bring together certain ideas about labeling and documentation in order to identify the SAS programming tools available to implement a meta-database strategy for documentation and retrieval.

The SAS* System makes available a wide variety of powerful programming tools, which can be combined to implement numerous problem solving strategies. These tools include labeling features, character string functions, SAS file manipulation, and user interface products, such as SAS/FSP* and SAS/AF*. This meta-database strategy is a synthesis of ideas for documentation and retrieval which can be implemented due to the flexibility of these tools.

No claim is made that this approach to creating meta-data or its partial implementation will solve the meta-data problem. It will be shown that the SAS System provides a simple yet powerful documentation capability due to the labeling features built into the SAS data step. The data about data stored in permanent SAS datasets and SAS format libraries can be combined with other information using SAS file manipulation capabilities. The output dataset options available with PROC CONTENTS and PROC FMTLIB are critical to this process. The INDEX function can then be used in the retrieval of patterns of existing information about data.

The meta-data must exist in a retrievable form. Detailed attention is given to methods for putting labeling information into permanent SAS datasets and SAS format libraries. It can then be manipulated and combined to prepare customized data dictionaries or other reports, as well as serve as the basis for the creation of meta-database(s). The menu creation and full screen data handling capabilities of SAS/AF and SAS/FSP software can make the documentation and retrieval system available to users who do not have sophisticated programming skills.

Meta-data Creation

The beginning of the meta-data creation and retrieval process is with a codebook. Codebooks are augmented record layouts about raw data files. They contain information about variables beyond the minimum needed to process raw data. The labeling information provides data processing decision makers with an understanding of the shorthand represented by variable names.

Labeling information from codebooks can be incorporated into the directories of SAS datasets or dictionaries of other system files through the use of programming statements. Means for efficiently creating SAS datasets from codebooks will be discussed in some detail. Machine-readable codebooks are the primary source of information about data. In a practical sense they are essential to the implementation of any meta-database strategy.

Codebooks do not always contain full sets of information about variables in a form which is easily obtained using the most efficient technique. This technique is to write a program to read the codebook as data and create SAS program lines. Sometimes it is easier to edit parts of a codebook with a word processor to generate the desired program lines. SAS uses the LABEL statement, FORMAT statement, ATTRIB statement, or PROC DATASETS to add label information to the directories of already existing SAS datasets.

PROC CONTENTS produces a consistently formatted report on the directories of SAS datasets. The report is, in effect, a codebook of a SAS dataset. Value labels are stored separately using PROC FORMAT and reported on using PROC FMTLIB. The information from PROC
CONTENTS and PROC FMTLIB would have to be combined in some way to create information about data usually found in a single codebook. All of the information contained within a SAS dataset and the accompanying SAS format library can be processed and output to a separate data dictionary file. The techniques will be described in this paper. One possible format for such a data dictionary file is the codebook template format. The format for the codebook template will be described later. It is one of several means of beginning to create a well-documented SAS dataset-SAS format library combination.

An edited codebook template results in SAS program lines for INPUT, LABEL, FORMAT, and PROC FORMAT statements. We can conclude the meta-database creation and retrieval process with a data dictionary of a form similar to that of the codebook template or expand it to contain additional information about variables included in SAS datasets of interest.

Codebook Template

A codebook template is a systematic arrangement of information contained in basic SAS statements used to create SAS datasets and SAS format libraries. It has been used by survey researchers to move useful information from their questionnaires to a codebook prior to entering their data. The systematic design of the codebook template helps in planning the INPUT statement and preparing label information according to SAS formatting syntax.

A text editor can be used to arrange information on variable names, variable labels, format names, and value labels into a file according to the codebook template. The codebook file can then be edited four times to set up the program lines for the INPUT, LABEL, FORMAT, and PROC FORMAT statements to be used in setting up a fully labeled SAS dataset - SAS format library combination. The editing could also be done using a word processing macro-facility or a set of SAS programs set up for this purpose.

A detailed presentation on the use of these programs was presented at SUGI 12. SAS/AF software can be used to set up a menu-driven application using this series of SAS programs. When additional SAS instructions are required, they can be included in the SAS program lines generated by this technique. The programs can then be submitted for the processing of raw data.

The syntax for the codebook template in its most simple form is presented in Figure 1. It can be seen that information for the INPUT and LABEL statements can be placed on the first line in the columns indicated. Just below the position for the LABEL statement is the information for the FORMAT statement. The information for the first occurrence of text needed for the VALUE statements used in PROC FORMAT can also be seen to begin in column twenty-five. It would be just below the name of the format associated with that set of value labels.

Figure 1 Basic Codebook Syntax

```
data entry columns
1      10      15     25
INPUT   LABEL   FORMAT
```

Only information for the INPUT statement is absolutely required when preparing a SAS program to read in raw data. Variable labels, formats and value labels are optional. Figure 2 contains the syntax for the codebook template with full labeling. Five possible combinations of acceptable syntax orderings for variables are demonstrated.

SAS syntax rules are followed in all instances. Both variable labels and value labels begin and end with single quotes. Format names are limited to eight characters ending with a period. The period can not be preceded by a number. The last value label to be associated with a format name must end in a semi-colon.

The value names used in VALUE statements in PROC FORMAT are created by removing the period from the SAS format names. Value labels included for a variable need not be repeated, if a
format name and similar set of value labels have already been defined. Only the format name for that variable needs to be indicated. It would be placed on the line below the information required for the INPUT statement. The last two lines in Figure 2 illustrate this option for the codebook template syntax.

Figure 2  Full Label Codebook Syntax

<table>
<thead>
<tr>
<th>data entry columns</th>
<th>1</th>
<th>10</th>
<th>15</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>type</td>
<td>informat</td>
<td>'var label'</td>
<td>format name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1='value label1'</td>
<td></td>
</tr>
</tbody>
</table>

Because it is arranged systematically and its syntax rules are well defined, a codebook template file can be produced by non-programmers. Translations can be made from codebooks in other formats or information can be selected from a questionnaire by a researcher. Due to the likelihood of typing mistakes, some SAS programming experience is usually needed to understand messages on the SASLOG generated when the SAS program lines from the edited program are executed.

The codebook template was designed to make it easier to create thoroughly labeled permanent SAS dataset and SAS format library combinations. It can be used with any operating system running Base SAS. Other options are available for creating labeled SAS programs. They will be discussed later.

Documentation Output from PROC CONTENTS

There are several reasons for preparing thoroughly labeled permanent SAS dataset - SAS format library combinations. After the initial effort required to prepare labeling information for SAS programs, users find that such combinations lead to significant saving in programming time and effort. The output from SAS procedures is now self-descriptive. Processing time is reduced. Dataset documentation is easily available for quick review using PROC CONTENTS and PROC FMTLIB. The SAS System can be used to help create meta-data because both PROC CONTENTS and PROC FMTLIB provide an output dataset option. This makes SAS labeling information accessible as data in the form of SAS datasets. These datasets can be combined in several ways to prepare reports or store information for later search and retrieval.

When an output dataset is created using PROC CONTENTS the variables from the original SAS dataset become observations in the CONTENTS dataset. This dataset will contain seventeen variables. A CONTENTS dataset can be merged with other datasets with similar observations. It is possible for survey researchers to add the text of questions to the CONTENTS datasets as new variables. Additional documentation on what were variables in the original SAS dataset can be added. Notes descriptive of variables are often included with census tapes. These notes can be added as the values of one or more new variables.

A list of key variables from a CONTENTS dataset is presented in Figure 3. The suggested additional variables could be, for example, QUESTEXTEXT and COMMENTS. NAME, LABEL, and FORMAT are listed from the regular set of variables output from PROC CONTENTS. Another regular variable mentioned is MEMLABEL. This is the name given by default to a SAS dataset label when it is present. This list of key variables is chosen because of their potential use with the SET and MERGE statements.

RETRIEVAL from CONTENTS Output Dataset

The key variables in the CONTENTS dataset will be subject to search. The INDEX function can be
used to search for character strings in any of the key variables. Variable names and the datasets where they are located can be identified. Researchers can then focus their attention to further diagnostics on variables in the original dataset(s). In this way researchers will be able to discover already collected information pertinent to their studies. This information will reduce the redundant collection of information and broaden the amount of information available to researchers testing various hypotheses.

Figure 3

Key Variables: CONTENTS

- MEMLABEL: dataset label
- NAME: variable name
- LABEL: variable label
- FORMAT: variable format

Suggested Additional Variables

- QUESTEXT: text of questionnaire
- COMMENTS: notes in comments file

One dataset in common use by social science researchers is the National Opinion Research Council (NORC) dataset. Data on 1157 variables has been collected and updated for the past fifteen years. This raw dataset was turned into a SAS dataset, which included variable labels. PROC CONTENTS was then run to create a CONTENTS dataset. A sample retrieval program was run to demonstrate one way of implementing this documentation and retrieval strategy.

The retrieval program is presented in Figure 4. After the CONTENTS dataset was produced a new dataset, RETRIEVE, was created. A SUBSETTING IF statement was used to extract only those observations which had the character string ‘race’ in the variable label.

Figure 4

Sample 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;

The SAS dataset RETRIEVE had seventeen observations. A researcher interested in research on ‘race’ could next go back to the NORC dataset and study the characteristics of the variables identified in this retrieval operation as being of possible interest. This simple example with real data is used to illustrate the potential for researchers of a meta-database system. Research consortiums, such as the International Consortium for Political and Social Research (ICPSR) have information about their data, which could be organized into a meta-database system using tools available with SAS software.

Large scale retrieval projects using multiple PROC CONTENTS output datasets are conceivable. Any number of CONTENTS datasets could be appended together using the SET statement. Where the storage and retrieval of such large datasets is cumbersome, it would be advisable to use a database management system. In line with relational database principles it would be advisable for each tuple (observation) to be uniquely identified on a primary key made up of a combination of the MEMLABEL variable and the NAME variable. The dataset could then efficiently use the indexed sequential access method (ISAM) storage structure. Searches could be done on the index variables, MEMLABEL and NAME. Updates to the database would be done sequentially at the end of the existing file.

Information about census data and other regularly generated government reports are candidates for this type of storage and retrieval process. The variable names used in several census studies are set up as
concatenated strings indicating information on sub-variables, such as geographic location and time period. This would further enhance the retrieval efficiency using some database management system software. SAS software could be used to create the meta-data. The DBMS could be used for storage and retrieval. SAS/ACCESS software could be used to create the appropriate SAS dataset resulting from a query of the database. SAS procedures could be used to report on the meta-data. If the original SAS datasets were accessible, further diagnostic processing could be done.

Documentation Output from PROC FMTLIB

Using PROC FMTLIB it is possible to create an output dataset from a permanent SAS format library. Fourteen variables are in such FMTLIB datasets. There is one observation for each value label. A FMTLIB dataset can be combined with other SAS datasets having similar observations.

Programmers might want to add variable name and dataset identifier information on variables using particular formats. A list of key variables from a FMTLIB dataset are presented in Figure 5. The additional variables suggested are MEMLOC and VARNAMES. FMTNAME, LABEL, and TYPE are variables regularly output from PROC FMTLIB. This list of key variables has been selected because of their potential use with the SET and MERGE statements.

Figure 5 Key Variables: FMTLIB

FMTNAME format module name
LABEL value label
TYPE type of format
Suggested Additional Variables
MEMLOC datasets where format used
VARNAMES variables using format

Retrieval from FMTLIB Output Dataset

The information about formats contained in the FMTLIB output dataset is of most use when merged with other information prior to search and retrieval. The CONTENTS output dataset and the FMTLIB output dataset contain a variable in common. This is the format name. Merges can be performed to rearrange information from these datasets so as to make the meta-data more accessible for search and retrieval. When the merged datasets are in existence, the INDEX function in SAS can be used as the basis for retrieving desired information from the SAS dataset. For large applications the appropriate storage structure and query design can be identified from among the possibilities available in database management systems.

A SAS dataset could be created which stores for later reference all variable names and SAS dataset labels for variables, using formats already stored in a format library. A check could be made to ensure that the value labels for the format referenced by a SAS dataset are the same as the value labels stored with the format module name in the format library. Once that is done, the variable name and dataset label could be merged to a FMTLIB output dataset. The format would be done by format name. The resulting expanded FMTLIB SAS dataset would identify variables from the same or different SAS datasets sharing the same format. This would provide researchers with an indication of where to look for variables which might be measures of information they can use in their research.

Another merge with great potential would include information from a FMTLIB output dataset with its accompanying CONTENTS dataset. The new dataset could be called "COMBINED". Each value label would be a separate variable in the combined SAS dataset and could be searched along with the other descriptive variables for desired character strings.
Data Dictionary Templates

The "COMBINED" SAS dataset will contain all of the information for each variable in a research SAS dataset. With this information it would be possible to recreate the program lines necessary to build a fully labeled SAS dataset - SAS format library combination. Included would be information on a variable's name, label, format name, value labels, variable type, position in the dataset, etc.

It would also be possible to prepare a report on this collection of 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. It, in turn, could be read by a properly prepared SAS program to create a set of SAS program lines in the same way as has been done with the codebook template approach. One possible design for such a standard report format would be a variation on the codebook template format.

Preparation of Labeled SAS Datasets

A variety of ways exist to create permanent fully labeled SAS datasets. Most SAS programs are written in a conventional way. Program lines are entered using a text editor or the Program Editor screen of the SAS Display Manager. The program lines for the creation of a permanent format library containing format modules to be used in conjunction with formats for variables in a SAS dataset are also, most often, created using conventional techniques. One alternative is the use of PROC FSEDIT. Another is the use of a text editor or SAS program to edit machine-readable codebooks. Attention will be given to various means for the efficient preparation of labeled SAS datasets and format libraries, because their efficient preparation is basic to the implementation of a meta-database strategy.

Rather than type in SAS program lines in a conventional way or use a variation of the codebook template, many SAS programmers use PROC FSEDIT to prepare labeled SAS datasets. The variable name, type, length, label, and format can be entered onto the screen provided by SAS with the DATA=NEW option of PROC FSEDIT. PROC FORMAT program lines must be prepared in a conventional way in order to create the value labels.

SAS datasets created with PROC FSEDIT do not include documentation of the history of the dataset creation in the same way as SAS datasets created using conventional program lines for the creation of SAS datasets. There are circumstances where it would be desirable to have this level of documentation. The program lines for datasets created using PROC FSEDIT could also be generated using a variation of the techniques being discussed here. While this is being done the order of variables could be rearranged using the LENGTH statement with the SET statement.

The print file created as output from PROC CONTENTS can be treated as a template for a codebook or data dictionary. It can be sent to a file, edited, and turned into SAS program lines. The data in the SAS dataset can be sent to a raw file. This raw file can then be read back in using the program lines just created. The result of these processes is the creation of SAS dataset with full documentation, including a history of the program lines used to create the dataset, and an indication of the source of the data.

In a similar way it is possible to convert other system files to SAS datasets. The data dictionary report can be sent to a file for editing. A SAS INPUT program can be created. Other statements can also be developed as is feasible. Most system files have utilities to write out raw data files. A SAS dataset can then be created including whatever labeling information can be transferred.

The record layouts or data dictionaries accompanying raw data files can also be edited so as to create a SAS INPUT program. Again, a text editor or specially written SAS program can be used to prepare SAS program lines. The source of raw data can be read in and a SAS dataset created.
SAS/ACCESS software provides a mechanism for creating SAS datasets directly from several database management systems. If the variable name in such a system is longer than eight characters, the DBMS variable name will be stored in the SAS dataset as a label.

DBMS programmers using systems for which SAS/ACCESS software is not yet available, who want to also use SAS, often create SAS INPUT statements from data definition information stored in system catalogs. They next unload the desired data from the DBMS and read it into a SAS dataset.

Levels of Meta-data

The National Opinion Research Council dataset (NORC) used in the example of how to use SAS tools for search and retrieval is typical of a large number of datasets used in the social sciences. It contains the results of a number of surveys, which are repeated on a yearly basis with only a few modifications. A new tape containing the results of these surveys is updated once a year. The data in raw form is arranged sequentially on the tape in such a way that researchers can easily process all or parts of it to produce a SAS dataset or other system file in the flat file format used most frequently by social scientists.

The NORC dataset in raw form is accompanied by a program to create a system file. Using methods previously described it is possible to edit the data dictionary of that system file to generate program lines to create a SAS dataset, which includes some labeling information.

The NORC dataset is a large example of a typical phenomena in the social sciences, a master dataset containing the results of numerous surveys. This is an efficient way to maintain the data, because there are few insertions, deletions, or updates to be made. It is, however, inefficient to store, because the dataset contains so many missing values.

If a file is typical of those required for an airlines reservation system or banking institution, where there are frequent insertions, deletions, and updates, the organization of the dataset in a format like the one described above would be unacceptable. It would be best in such cases to use normalization techniques, such as are used with DBMS databases. In such circumstances the programmer should decompose the master SAS dataset into subsets for data management purposes. The master dataset could be reconstructed by means of file manipulation tools, such as the MERGE and SET statements for documentation and analysis purposes.

Where problems do not exist with insertions, deletions, and updates to a dataset, researchers utilizing multiple project specific surveys for a particular study might want to combine all of the information on their subjects into a master dataset using SAS. Experience has shown that preliminary labeled SAS dataset and format library combinations for each survey can be prepared separately. It is best to use techniques, which directly result in the necessary SAS program lines. With an extra step SAS datasets created through PROC FSEDIT can indirectly yield such program lines. A master SAS dataset can then be created by combining the necessary program lines into one file ending with the appropriate MERGE statement. In a similar manner a SAS programmer could combine the program lines for a particular study into one program to create a single format library for that study. If careful attention is given in this process to the naming and labeling of variables, an efficient means can be developed to set up a system for longitudinal, as well as one time studies.

The resulting master SAS dataset will have in one place the programming documentation for the creation of the dataset. Included will be a record of the coding for the creation of all new variables. PROC CONTENTS with the HISTORY option will write such a report. It will also contain the makings of a meta-database for the study. The meta-database can be shared with other researchers as desired.

Federal agencies, such as the Bureau of the Census and the National Technical Information Service, and special social and economic research organizations, which include the Roper Center and the International Consortium for Political and Social Research, distribute data files containing information on thousands of variables. These datasets are usually accompanied by data dictionary files or other machine-readable sources...
of information about the data. The datasets are
sometimes accompanied by SAS programs to read
in the data. Systematically accessing detailed
information about the variables in these datasets
is not possible. The meta-database strategy
described in this paper could be used to gather
together the available information and make it
retrievable.

**SAS User Interface Tools**

PROC DATASETS or the ATTRIB statement
can be used to add labels and formats for
variables. PROC FORMAT can be used to
create the value labels. These procedures
and SAS statements are the tools to be used
to complete the labeling of SAS datasets
initially created with only partial labeling.

PROC DATASETS can be used, either through
the creation of SAS program lines or
interactively from the SAS Display Manager
to add variable label and format information
to SAS datasets. To learn the interactive
use of PROC DATASETS it would be helpful
to review the SASCBT course, "Managing
Your SAS’ Data Library”.

SAS/AF menus can be used in conjunction
with the implementation of this
meta-database strategy. Help files can be
developed to guide programmers through
various steps in the strategy. The series of
programs used in the codebook template
approach and a similar series of programs
using SAS/FSP software can be incorporated
into SAS/AF menu driven applications. Such
is also the case with other elements of the
meta-database strategy.

**Summary**

The diagram in Appendix I is intended to begin to
illustrate the way tools available within the SAS
system can be used to implement a meta-database
strategy for documentation and retrieval. It can
be seen that the strategy is composed of a
synthesis of ideas made possible through the use
of several combinations of SAS tools.

Documentation is initially made possible through
the use of labeling features built into the SAS
system. Labeled SAS programs include dataset
labels and variable labels in a permanent SAS
dataset and are connected through the FORMAT
statement to the value labels in a permanent SAS
format library. There are several ways of creating
documented SAS datasets. The labeling
information included with machine-readable
codebooks accompanying large research datasets
or government generated data is seen as a key
source of information on data, which could be
included in general purpose meta-data systems.
Researchers can use several programming ideas
including the codebook template approach to
organize information about variables and create
meta-databases for particular multi-survey
studies. The ATTRIB statement and PROC
DATASETS are seen as tools to enhance labeling
for existing SAS datasets.

Documentation can be made accessible through the
output dataset options available with PROC
CONTENTS and PROC FMTLIB. Additional
information can be added to these output datasets
for later search and retrieval. Retrieval can be
done using tools, such as the INDEX function, which
are part of the Base SAS product. For large scale
storage and retrieval tasks database management
systems, such as SYSTEM 2000® would be most
efficient.

Using SAS file manipulation tools it is possible to
merge information from the CONTENTS and FMTLIB
output SAS datasets. The resulting COMBINED SAS
dataset potentially contains all of the information
needed to implement the meta-database strategy.
Variables in the COMBINED SAS dataset can be
searched and variables in research datasets
identified for later examination. It is also
possible to generate reports in a data dictionary
template format using the meta-database.

Academic research applications of this
meta-database strategy have been emphasised, but
government and business information systems
could also put this strategy to use. SAS/AF could
be used to make menu-driven applications of the
strategy possible for non-programmers.
REFERENCES


Acknowledgements

This paper would not have been possible without the support of my friends and co-workers in the User Services Division of the University Computing Center at the University of Rochester.

For more information contact the author at the following address:

University Computing Center - Taylor Hall,
University of Rochester, Rochester, NY 14627.

SAS*, System 2000*, SAS/FSP* and SAS/AF* are registered trademarks or trademarks of SAS Institute, Inc.
Appendix I Model of Meta-database Implementation Strategy for Documentation and Retrieval Using the SAS* System