Taking an HTML Snapshot of a Data Warehouse: Just say, Cheese!

Eric Simms, Linden International, Wayne, PA
Jerry Kagan, Kagan Associates, Inc., Devon, PA

Abstract

This paper will discuss how the New Hampshire Department of Health and Human Services implemented a system using SAS and web technology to validate and document data elements within their data warehouse. This paper is significant because the general approach taken allows the techniques used to be applied to any SAS based data warehouse. Furthermore, the use of web technology and the programmatic creation of HTML pages allows for instantaneous distribution and insures the accuracy of the information.

Introduction

Typically, a data warehouse is not started from scratch, but rather is created by collecting existing data from legacy systems. Often, the only people with any knowledge of the data elements in these legacy systems are non-technical users. The assistance and good-will of these users is crucial to the success of the data warehouse project.

The difficult task of collecting data from these existing systems also has the great misfortune of occurring before any of the benefits of a data warehouse are apparent to the user community. It is exactly at this time that the developers will be asking the user community to contribute their valuable time and knowledge in helping to get the warehouse started. Therefore, any utility which simplifies the process and additionally allows some of the potential of a data warehouse to peek through can be of great benefit.

The Snapshot utility that is described below is an example of how web technology was used effectively to get all people with a stake in the data warehouse actively and effectively involved in its creation.

The People

Except for the one-man shops, it takes a team to build a warehouse. Some warehouse projects can involve many different programmers, each bringing data from portions of different legacy systems, and a collection of users who are experts concerning their data. Quite often those experts are the only source of information about the data due to outdated or misleading documentation. These experts will most likely not interrelate in their normal day-to-day work, but they will all need equal access to the metadata in order to identify common data elements, relationships and problems.

The Process

As the authors have experienced it over the years, the process looks like this:

A SAS programmer, lets say it's you, is asked by a user representative to get some existing data into a SAS data set. You ask many questions about the data, one of which is "Where is the data now?" If the user-rep can't answer that, then you go back to learning JAVA in hopes of a future career move. Otherwise you ask, hopefully, "Is there a file layout of the data? Perhaps a COBOL program which wrote the data to begin with?" If this is provided, you are happy, If not, you whine but make an attempt. In either case you try reading it into SAS until you get rid of any errors in the log. You then show what you have to the user-rep who makes a superficial examination and requests changes, you re-run. The user-rep finally takes a closer look at the data contents, offers yet more comments, causing more re-runs, etc.

The Data

Whether using a utility such as COB2SAS[1] or writing input statements, the existing data must be read from its location and format into a SAS data set or view. At this time the basic characteristics of each data element must be deduced i.e.: character or numeric, length, format and label.

At the same time, the warehouse developers are also (possibly for the first time) trying to implement standards within and across systems. Any element which occurs on multiple tables must have the same characteristics throughout the warehouse. This process usually takes multiple iterations to complete. Adding further complications, it is crucial to be able to provide this information to all the parties involved and rapidly refresh the information as it is corrected.

[1] COB2SAS is a sample program provided by SAS Institute Inc. as a tool that can assist you in converting COBOL language data description entries into equivalent SAS language statements

Format standards and naming conventions need to be clearly defined. Some examples of problems caused by the absence of standards include variables with the same name that represent different things, and variables that represent the same thing with different names, descriptions, formats or unit measures. For instance, a variable may be named SEX in one table with values of 1 and 2 and be named GENDER with values of M and F in another. Date formats could vary causing some reports to display dates in ddMonyy format while others use yy/mm/dd format, and yet others dd/mm/yy format. Another common problem is financial numbers reported in thousands in one table and millions in another.

Typically the data relationships are more complicated than the examples above and problems only come to light after much effort. In any event, these types of issues must be brought to the table and resolved early in the process of acquiring the data. Any delay will result in additional work as any programs/systems written making use of the non-standard (unclean) data will eventually have to be modified in order to conform to standards.

A Solution - The HTML Snapshot

Using a combination of tools consisting of the new SAS HTML formatting macros (SAS2HTML, TAB2HTML, and OUT2HTML)[2] and the SAS Automation Plugin[3], along with Base SAS software and the information available in the SAS dictionary tables, the authors created a utility to generate an HTML snapshot of any SAS based data warehouse.

The HTML is generated by a relatively simple SAS program which captures the needed information from the dictionary tables and generates HTML by use of PUT statements and the HTML formatting macros. This snapshot contains all of the characteristics for each table (name, label, number of observations, etc.), and the characteristics of each variable within each table. Assorted cross-referenced pages and exception reports are produced at the same time which are used to identify problems and typos which could later cause problems in warehouse applications. This utility makes the warehouse metadata (and optionally the data) available so that any user with a web browser will be able to examine the information.

Dictionary Tables

Dictionary tables hold metadata on all SAS tables and have been available since version 6.07 of the SAS System. They are created automatically at run time and are accessible from PROC SQL using the reserved libname DICTIONARY, or by using views defined in libname SASHELP. (Dilorio & Michal, SUGI 21 Proceedings, p.213)

Table 1: Dictionary Tables used in Snapshot

SASHELP.VTABLE	Information about tables within all libraries
SASHELP.VCOLUMN	Information about variables within tables
SASHELP.VINDEX	Table indexing information
SASHELP.VSLIB	List of libnames and paths

Security Concerns

It should be noted that if a data set is password protected then it won't show up in the dictionary columns. The authors solved this problem by creating views for all such tables and specifying obs=0 in the view. As a result the data can be confidential, but the information ABOUT the data is open to all. If your organization has any concern or hesitancy over this, the authors have found it helpful to explain it in this manner:

"While someone's SSN, etc. may be considered confidential, the fact that we are storing these fields in our warehouse should not be (unless we are the CIA). If anyone is concerned about people knowing that we store some field then it should be a red-flag and brought to the highest levels for a policy decision. If we are storing data that we should not be then perhaps the president of the company would like to know about this. Preferably from us, and not Mike Wallace. Tick. Tick. Tick..."

Providing Data Contents and Frequency Tables

By making use of the SAS Automation Plugin, which allows a link in the HTML code to start a local SAS session and specify a particular SAS program to execute, the users with SAS on their desktops can examine the data interactively. Two examples of this are (1) Clicking on a table name presents the user with a VIEWTABLE of the table; and (2) Clicking on a variable name presents the user with the output of a PROC FREQ on said variable.

[2] The HTML formatting macros are available for free download on the SAS Institute Web site at www.sas.com

[3] The SAS Automation Plugin is available for free download at www.sas.com and enables you to build Web pages that include a SAS action.

While this is a great benefit to the warehouse builders in helping to clean up the data contents and define some limits (i.e. a certain field should not contain more than $1000, and if it does we need to produce an exception report), it also allows early access to the warehouse for the users. In some cases, it may be the first time that they can examine their data without having to go through the IT department.

Maintenance/Documentation Benefit

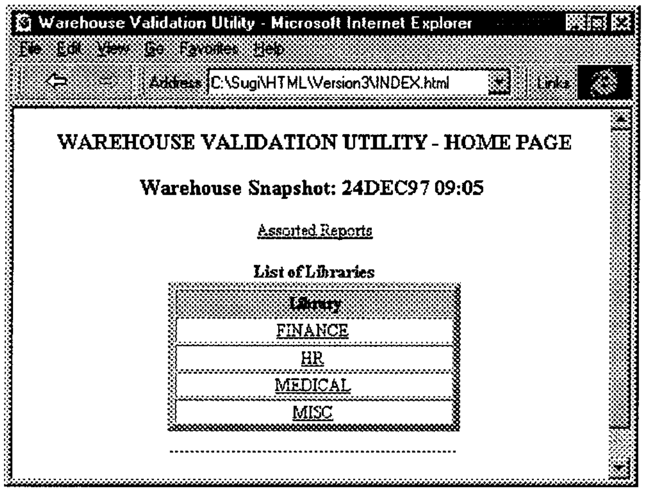
As systems are brought into the warehouse, the Snapshot utility will be providing documentation of all the tables and data elements. This documentation can easily be kept up to date and delivered throughout the organization. In addition, as new systems are brought in, we can ensure that their data elements are in accordance with existing standards i.e.: If variable SEX has a label of GENDER and is of type CHARACTER (e.g. M or F), then a new field of SEX with type NUMERIC (e.g. 1 or 2) will show up on an exception report and will be easily identified and corrected.

The Web Screens

Figure 1 shows the first page, or home page, of the web site created by the Snapshot utility. This provides the user with a link to *Assorted Reports* and a table of links to available libraries.

Selecting the FINANCE library displays a table of the data sets and views available in that library along with descriptions, number of variables, number of observations and whether or not the data set is indexed (figure 2).

Figure 1: The Home Page


Warehouse Validation Utility - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address C:\Sugi\HTML\Version3\INDEX.html
Links

WAREHOUSE VALIDATION UTILITY - HOME PAGE

Warehouse Snapshot: 24DEC97 09:05

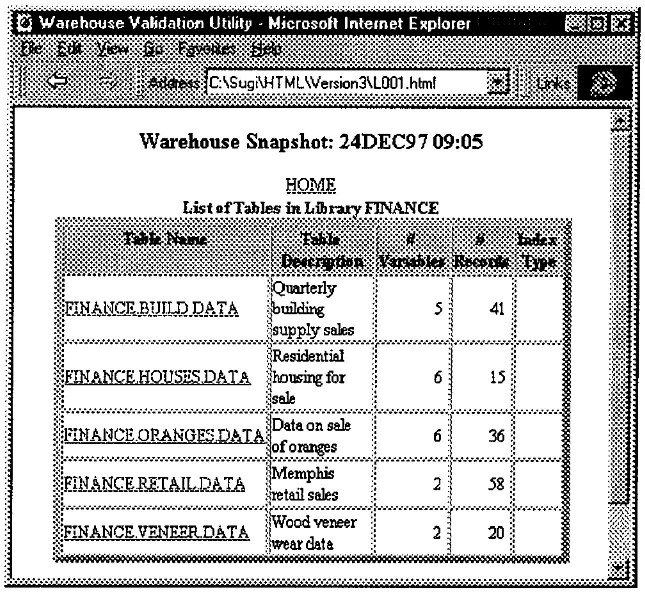
Assorted Reports

List of Libraries

Library
FINANCE
HR
MEDICAL
MISC

Selecting the BUILD data set in the Finance library shows a table of information about the variables in that data set and a button labeled *View Data,* which calls the SAS Automation Plugin (figure 3). If the user has SAS installed, then clicking this button will start a SAS session on their PC and will execute the code specified in the EMBED tag in the HTML (Tables 4 & 5).

Figure 2: Tables within Finance Library


Warehouse Validation Utility - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address C:\Sugi\HTML\Version3\L001.html
Links

Warehouse Snapshot: 24DEC97 09:05

HOME

List of Tables in Library FINANCE

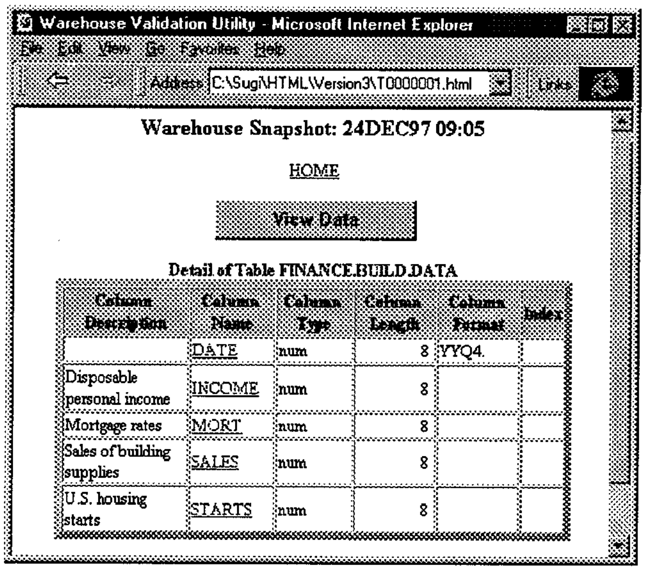
Table Name	Table Description	# Variables	# Records	Index Type
FINANCE.BUILD.DATA	Quarterly building supply sales	5	41	
FINANCE.HOUSES.DATA	Residential housing for sale	6	15	
FINANCE.ORANGES.DATA	Data on sale of oranges	6	36	
FINANCE.RETAIL.DATA	Memphis retail sales	2	58	
FINANCE.VENEER.DATA	Wood veneer wear data	2	20	

In this case the users will be dropped into a VIEWTABLE of the data set (figure 8), where they can then browse the data, move and hide columns, set WHERE clauses, print, export to Excel, etc. In the production version SAS/CONNECT® is used to connect to a warehouse located on a mainframe. This has the advantage of inheriting the security already in place on the mainframe, as the user is prompted for their user id and password.

Note that in figure 3 the variable DATE does not have a label. Not only is this a missed opportunity to get some free documentation (SAS will propagate the information automatically) but it may instigate problems down the road if any applications rely on being able to access a meaningful label. This type of knowledge is so important that separate reports are generated to make this information available without a lot of cross checking (See figure 5).

Selecting the SALES variable from the BUILD data set shows us some of what we already know (figure 4), but also tells us that SALES is also available in the FINANCE.RETAIL data set. This is of interest to programmers trying to understand the data or looking for merge keys. However, it is clear from the labels that these two variables named SALES are in fact two entirely different things. Perhaps names of SALESB (building) and SALESR (retail) would be more appropriate. This is easily discernible because these variables have meaningful labels, otherwise this information would have slipped by us only to be known by the "Keepers of the Company Folklore."

Figure 3: Variables within Table


Warehouse Validation Utility - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address C:\Sugi\HTML\Version3\T0000001.html
Links
HOME
View Data

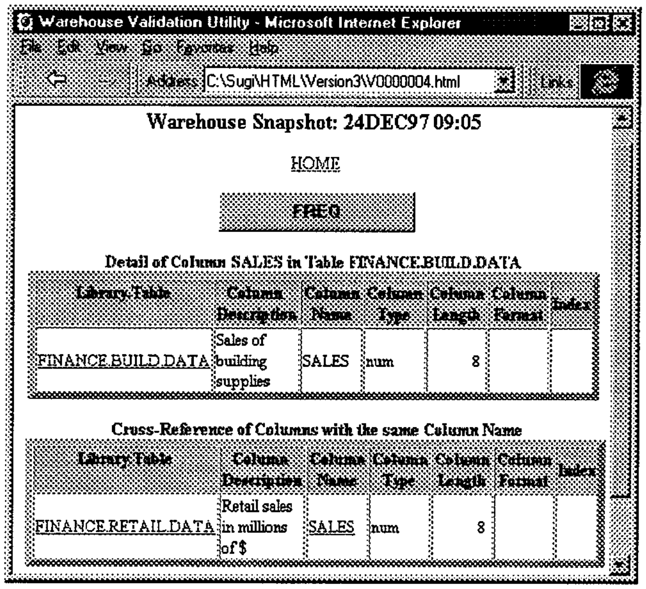
Warehouse Snapshot: 24DEC97 09:05

Detail of Table FINANCE.BUILD.DATA

Column Description	Column Name	Column Type	Column Length	Column Format	Index
	DATE	num	8	YYQ4.	
Disposable personal income	INCOME	num	8		
Mortgage rates	MORT	num	8		
Sales of building supplies	SALES	num	8		
U.S. housing starts	STARTS	num	8		

The button labeled FREQ (figure 4) is another use of the Automation Plugin. In this case it runs a frequency distribution on the selected variable. The user should be warned that it is probably not a good idea to run a frequency distribution on any non-discreet field, for example a dollar figure.

Figure 4: Details of Variable


Warehouse Validation Utility - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address C:\Sugi\HTML\Version3\V0000004.html
Links
HOME
FREQ

Warehouse Snapshot: 24DEC97 09:05

Detail of Column SALES in Table FINANCE.BUILD.DATA

Library.Table	Column Description	Column Name	Column Type	Column Length	Column Format	Index
FINANCE.BUILD.DATA	Sales of building supplies	SALES	num	8		

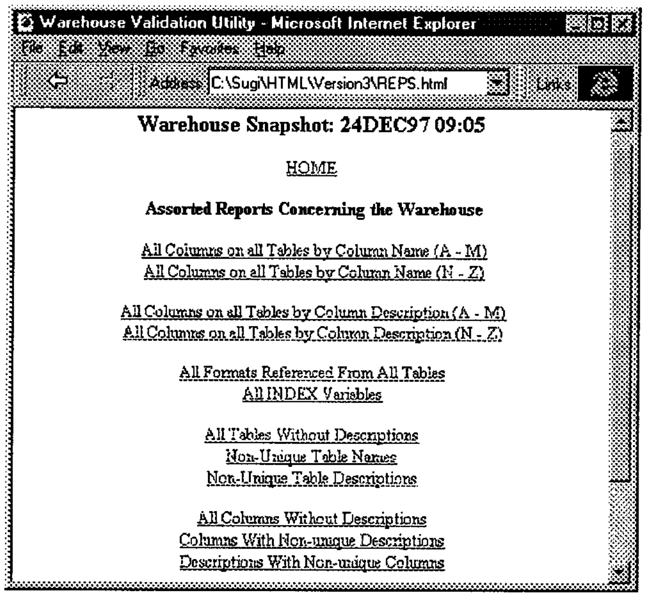
Cross-Reference of Columns with the same Column Name

Library.Table	Column Description	Column Name	Column Type	Column Length	Column Format	Index
FINANCE.RETAIL.DATA	Retail sales in millions of $	SALES	num	8		

Use of the Plugin allows the next step in warehouse building to take place—the step from metadata validation to data validation. A side benefit is that the users can generate simple reports for themselves. In fact, these plugins may provide the first deliverables from the warehouse, and an easy path to the data without the usual battle with IT for resources.

Figure 5 shows a list of reports generated for use in validation. Figures 6 and 7 show examples of two of these reports. Note that in the list of all columns by column description, the variables without descriptions are listed first.

Figure 5: Available Reports


Warehouse Validation Utility - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address C:\Sugi\HTML\Version3\REPS.html
Links
HOME

Warehouse Snapshot: 24DEC97 09:05

Assorted Reports Concerning the Warehouse

All Columns on all Tables by Column Name (A - M)
All Columns on all Tables by Column Name (N - Z)

All Columns on all Tables by Column Description (A - M)
All Columns on all Tables by Column Description (N - Z)

All Formats Referenced From All Tables
All INDEX Variables

All Tables Without Descriptions
Non-Unique Table Names
Non-Unique Table Descriptions

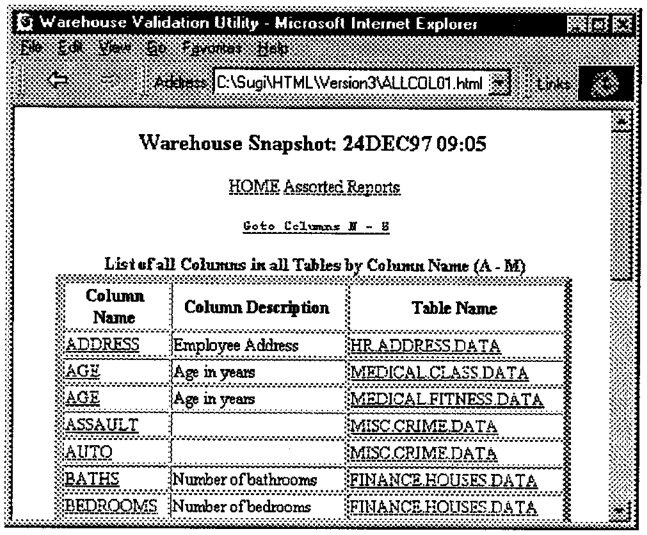
All Columns Without Descriptions
Columns With Non-unique Descriptions
Descriptions With Non-unique Columns

Static vs. Dynamic Content

The first version of the Snapshot application produced a series of static inter-linked HTML documents. This worked well in the beginning as a new snapshot was taken each night when the warehouse was updated. However, there comes a time when a particular warehouse will become too big to do a static snapshot gracefully. Currently, there are over 6000 elements and over 350 tables in the DHHS warehouse, and it is still in the early stages. Generating the many thousands of static HTML files needed to track these elements has become difficult to accomplish in a timely manner. The next step was to take advantage of SAS/IntrNet® in order to generate most of these pages dynamically.

Documenting this conversion is beyond the scope of this paper, however, it was not difficult to convert the code that created the static HTML to be used for the dynamic application version, and visually the change was transparent to the users. Briefly, the logic used to generate the unique HTML document names was removed and instead of writing output to a file it is written directly to the user's web browser (_WEBOUT). In the HTML generated, instead of hypertext links to static HTML pages, the SAS programs are called directly with specific parameters.

Figure 6: All Variables by Name



Warehouse Validation Utility - Microsoft Internet Explorer

Address C:\Sugi\HTML\Version3\ALLCOL01.html

Warehouse Snapshot: 24DEC97 09:05

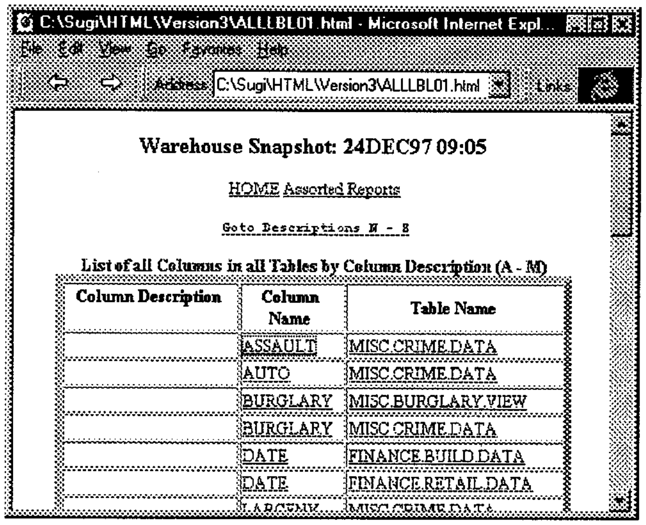
HOME Assorted Reports

Goto Columns N - Z

List of all Columns in all Tables by Column Name (A - M)

Column Name	Column Description	Table Name
ADDRESS	Employee Address	HR.ADDRESS.DATA
AGE	Age in years	MEDICAL.CLASS.DATA
AGE	Age in years	MEDICAL.FITNESS.DATA
ASSAULT		MISC.CRIME.DATA
AUTO		MISC.CRIME.DATA
BATHS	Number of bathrooms	FINANCE.HOUSES.DATA
BEDROOMS	Number of bedrooms	FINANCE.HOUSES.DATA

Figure 7: All Variables by Label



C:\Sugi\HTML\Version3\ALLLBL01.html - Microsoft Internet Expl...

Address C:\Sugi\HTML\Version3\ALLLBL01.html

Warehouse Snapshot: 24DEC97 09:05

HOME Assorted Reports

Goto Descriptions N - Z

List of all Columns in all Tables by Column Description (A - M)

Column Description	Column Name	Table Name
	ASSAULT	MISC.CRIME.DATA
	AUTO	MISC.CRIME.DATA
	BURGLARY	MISC.BURGLARY.VIEW
	BURGLARY	MISC.CRIME.DATA
	DATE	FINANCE.BUILD.DATA
	DATE	FINANCE.RETAIL.DATA
	LARCENY	MISC.CRIME.DATA

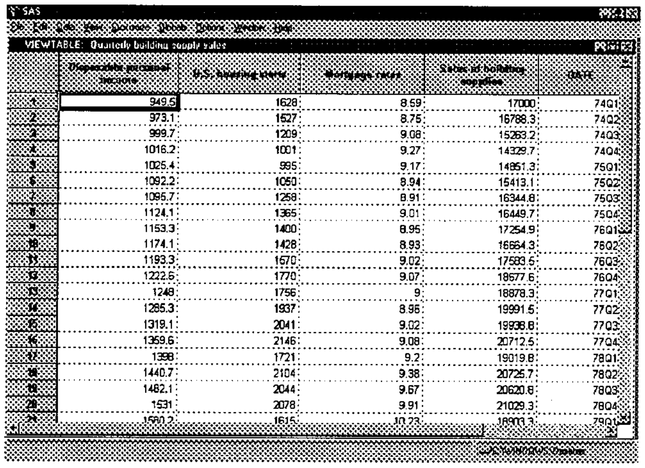
The SAS Code

While, most of the programming used in the Snapshot utility consists of basic SAS elements there are a few techniques used that may not be completely obvious; using the dictionary tables to access the metadata, writing multiple output files in one data step, using the SAS formatting macros, and using the SAS Automation Plugin.

The first step was to get the needed metadata into temporary data sets. Table 2 gives one example of how the metadata is accessed using the dictionary tables. This example shows how a table of information about tables is created. Note the macro reference %libs--this macro is not shown but simply includes the names of previously defined librefs. The dictionary tables provide all the input for the snapshot utility; the HTML pages are the output.

A combination of two techniques are used to generate the HTML pages; DATA _NULL_ steps and the SAS formatting macros. The main page and all index pages are created with DATA _NULL_ steps (figures 1-5). Using the tables of metadata on libraries, tables in libraries and variables in tables the application spins through each data set one record at a time and writes out HTML to a particular file for that record. Table 3 illustrates this technique.

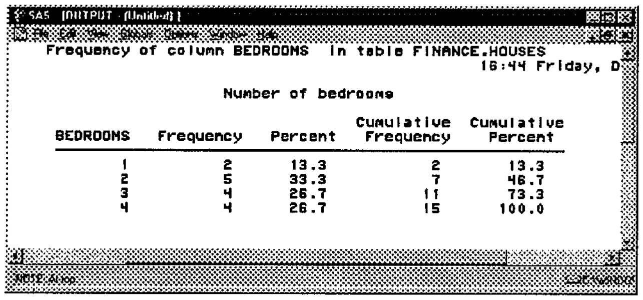
Figure 8: Sample Result of View Data Function



VIEWTABLE: Quarterly building supply sales

	Disposable personal income	U.S. housing starts	Mortgage rates	Sales of building supplies	DATE
1	949.5	1628	8.59	17000	74Q1
2	973.1	1527	8.75	16788.3	74Q2
3	999.7	1209	9.08	15263.2	74Q3
4	1016.2	1001	9.27	14329.7	74Q4
5	1025.4	995	9.17	14851.3	75Q1
6	1092.2	1050	8.94	15413.1	75Q2
7	1095.7	1258	8.91	16344.8	75Q3
8	1124.1	1365	9.01	16449.7	75Q4
9	1153.3	1400	8.95	17254.9	76Q1
10	1174.1	1428	8.93	16664.3	76Q2
11	1193.3	1570	9.02	17583.5	76Q3
12	1222.6	1770	9.07	18677.6	76Q4
13	1248	1756	9	18878.3	77Q1
14	1285.3	1937	8.96	19991.5	77Q2
15	1319.1	2041	9.02	19938.8	77Q3
16	1369.6	2146	9.08	20712.5	77Q4
17	1398	1721	9.2	19019.8	78Q1
18	1440.7	2104	9.38	20725.7	78Q2
19	1482.1	2044	9.67	20620.8	78Q3
20	1531	2078	9.91	21029.3	78Q4
21	1580.2	1615	10.23	18903.3	79Q1

Figure 9: Sample Result of FREQ Function



SAS OUTPUT - (Untitled)

```
  Frequency of column BEDROOMS  In table FINANCE.HOUSES
                                           16:44 Friday, D

                       Number of bedrooms

                                            Cumulative   Cumulative
BEDROOMS    Frequency    Percent    Frequency      Percent
-----------------------------------------------------------
       1            2       13.3            2         13.3
       2            5       33.3            7         46.7
       3            4       26.7           11         73.3
       4            4       26.7           15        100.0
```

The assorted reports (figures 6-7) are generated using the %DS2HTM macro with HTML tags embedded into the data. The macro parameter ENCODE=NO is needed to prevent the tags from being encoded (commented out) by the macro and DATA _NULL_ steps before and after the call are used to output additional HTML. To maintain a consistent look and feel, macros are used to write the header and footer of each page.

Table 2: Accessing the Metadata

```
*** Get metadata about tables;
data metadata.tables (keep=libname memname memtype memlabel
                              obscount nvar indxtype);
  set sashelp.vtable
    (where = (libname in (%libs)
     and memtype in ('DATA','VIEW')));
  label obscount = 'Number of records';
  if nobs ge 0 then obscount = nobs - delobs;
run;
```

Table 3: Multiple output files per data step

```
*** Assign unique links for each library and table;
data tables;
  set metadata.tables;
  by libname;
  . . .
  if first.libname then do;
    count+1;

    *** e.g.: L0000001 ... Lnnnnnnn;
    linkName = 'L' || put(count, z7.);

    *** e.g.: <A href="L0000001.html">FINANCE</A>;
    libTag ='<A href="'||trim(linkName)
          ||'.html">'||trim(libname)||'</A>';
  end;

  *** e.g.: FINANCE.BUILD.DATA;
  libTable  = trim(libname)||'.'||trim(memname)||'.'||memtype;

  *** e.g.: T0000001 ... Tnnnnnnn;
  tableNam = 'T' || put(_n_,z7.);

  *** e.g.: <A href="T0000001.html">FINANCE.BUILD.DATA</A>;
  tableTag = '<A href="'||tableNam||'.html">'||trim(libTable)||'</A>';
  . . .
run;

*** Generate the many HTML files;
data _NULL_;
  length lib_dd $80;`
  set tables;
  by libname;
  . . .
  *** e.g.: c:\sugi\whouse\L0000001.html;
  lib_dd = "&pathnm."||linkName||'.html';

  *** When filevar changes the current file closes and a new one;
  *** automatically opens;
  file libhtml filevar=lib_dd;

  if first.libname then
  do;
    WRITE OUT A LINK FOR EACH TABLE IN THE
    LIBRARY, ETC.
  end;
  . . .
```

Table 4: SAS Automation Plugin Sample HTML

```
<!-- Sample HTML to Display View Table Button -->
<EMBED SRC=" T0000001.sas" label="View Data"
height=30 width=150 action=Submit datatype=SASFile>
```

Table 5: SAS Automation Plugin Sample SAS Code

```
*** From file T0000001.sas;
*** dm statement tries to hide internal workings of SAS from user;
dm 'command close;toolclose;pgm;icon on;output;icon on;log;icon
on';
libname FINANCE 'c:\sugi\whouse\finance';
dm 'awsmaximize;zoom;viewtable FINANCE.BUILD ' continue;
run;
```

Table 6: Output File Naming Conventions

INDEX.HTML	Home page
Lnnnnnnn.HTML	Tables within each library
REPORTS.HTML	Index of various reports
Tnnnnnnn.HTML	Details of each table
Tnnnnnnn.SAS	Displays View Table
Vnnnnnnn.HTML	Details of each variable
Vnnnnnnn.SAS	Displays proc freq of variable
ALLxxxxx.HTML	Various Reports

Conclusion

By making use of the Snapshot utility the users are able to see what the DHHS warehouse truly looks like--not what it used to look like, but what it really is. Users can view the metadata from any location on the organization's intra-net with only a web browser. Even when a data dictionary has been produced (usually a binder full of printouts) programmers don't use it because they know that it is obsolete. Instead, they will run some code, e.g. a PROC CONTENTS, to get a true picture of the data. No matter what is used as a data dictionary, 95% of the effort is in getting the information delivered to those who need it, and in getting someone with the business knowledge and familiarity with the data to review it. The Snapshot utility eloquently addresses these issues.

Contacts

The authors welcome any questions or comments.

Eric Simms, bd813@freenet.carleton.ca
Jerry Kagan, JerryKagan@msn.com

References & Acknowledgments

The authors wish to thank Glen Smith and Alan Dickson of the New Hampshire Department of Health and Human Services for their generous help.

"Data About Data: An Introduction to Dictionary Tables," Dilorio & Michal, SUGI 21 Proceedings