Extracting SAS Data Sets From a Multidimensional Database in Batch


ABSTRACT

In Release 6.12, Multidimensional database (MDDB) files are a new type of SAS file that stores summarized data for fast and easy access. SAS/EIS® software offers several objects that can interactively extract data from the MDDB into SAS data sets. How do you perform this operation in batch for delivery of summarized data to other processing systems? A SAS/AF® class exists that can be used to extract selections from a MDDB file. This paper demonstrates how to extract data from a MDDB file using a SAS/AF SCL entry that can run in a batch process to deliver your aggregate data to the rest of the SAS system.

INTRODUCTION

The Economic Area of the Census Bureau conducts the Census of Manufacturing and Minerals Industries (CMMI) every five years. I investigated the use of SAS 6.12 Online Analytical Processing (OLAP) tools to create data review systems for the analysts. The tools had to satisfy the following requirements:

1. Summarize all levels of aggregation in a window of six hours or less.
2. Provide a tool to allow the analysts to review any level of aggregation on demand.
3. Extract the aggregations for input to other processing systems.

The SAS/MDDB Server™, SAS/EIS and SAS/AF software satisfied these requirements.

Bureau analysts review the CMMI data in two phases. Phase one is an industry series review; tabulations on active establishments at various levels of an industry code and limited levels of geography (i.e., state). The support staff use the OLAP tools, the SAS/MDDB Server and the SAS/EIS Multidimensional Report objects for phase one. The batch programs, run at night, extract data from a DBMS that create the detail SAS data set that serves as input to create the MDDB. Analysts review the data contained in the MDDB with SAS/EIS Multidimensional reports.

Phase two, disclosure, begins when phase one is completed. Analysts review the data, applying disclosure (table cell suppression) to the data before table publication. This is an interactive SAS/AF system. Disclosure requires aggregations from phase one as input for the table cell suppression algorithms. Since the MDDB contains those aggregations, we developed a way to extract those summarizations in a batch process. This paper demonstrates the approach using an example MDDB and SAS/AF Screen Control Language (SCL) source code to obtain these aggregations and store them in a SAS data set.

The data used in this paper has been randomly generated and simulates characteristics of the original survey data. The original data is under Title 13 confidentiality restrictions and cannot be used.

MDDB Creation

For this paper, the detail SAS data set used to create the MDDB is a one percent sample (5,000 observations) with seven categorical variables and seven analysis variables. The original SAS data set contains approximately 500,000 observations, 12 categorical variables and approximately ninety analysis variables. The following describes the variables in the sample SAS data set.

<table>
<thead>
<tr>
<th>Categorical Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>COID</td>
<td>Company ID</td>
</tr>
<tr>
<td>IND6,4,3,2</td>
<td>Industry code classification variable for N-digits</td>
</tr>
<tr>
<td>COSZ</td>
<td>Company Category Size</td>
</tr>
<tr>
<td>ST</td>
<td>State Code</td>
</tr>
<tr>
<td>Analysis Variable</td>
<td>Definition</td>
</tr>
<tr>
<td>EST</td>
<td>Number of establishments that comprise a company</td>
</tr>
<tr>
<td>TE</td>
<td>Total Employment</td>
</tr>
<tr>
<td>SW</td>
<td>Salary and Wages</td>
</tr>
<tr>
<td>CO6,4,3,2</td>
<td>Company count variable for each N-digit Industry Level</td>
</tr>
</tbody>
</table>

The MDDB created from the sample SAS data set consists of the following:
Categorical
Variables

<table>
<thead>
<tr>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND6</td>
</tr>
<tr>
<td>IND4</td>
</tr>
<tr>
<td>IND3</td>
</tr>
<tr>
<td>IND2</td>
</tr>
<tr>
<td>ST</td>
</tr>
<tr>
<td>COSZ</td>
</tr>
</tbody>
</table>

Hierarchy

<table>
<thead>
<tr>
<th>Cells</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWAY</td>
<td>IND6 IND4 IND3 IND2 ST COSZ</td>
</tr>
<tr>
<td>HIER1</td>
<td>IND6 IND4 IND3 IND2</td>
</tr>
<tr>
<td>HIER2</td>
<td>ST</td>
</tr>
</tbody>
</table>

The SUM statistic was requested for all analysis variables defined in the sample SAS data set.

MDDB Extraction Process

Once the MDDB is created, we extract the levels of aggregation for the disclosure review process. Note, the SAS/EIS Multidimensional Report Object allows you to interactively extract any level of aggregation to a SAS data set by selecting the following from the pull-down menus in the report:

**File > Save As... > SAS data set**

Because the MDDB creation occurs at night, we needed a process to perform this extraction automatically.

PROC SUMMARY/MEANS can summarize the detail SAS data set at the various levels of aggregation for the disclosure process. The disadvantage to this approach is that it requires redundant processing of the large detail SAS data set expending additional computing resources for this process.

The solution is that a SAS/AF class performs as an application program interface (API) to the MDDB. The MDDB_M class and SCL can perform the extract. PROC DISPLAY executes the code in the SCL entry in batch. Figure 1 is the PROC DISPLAY code needed to execute the SCL entry.

**The SCL Entry**

A SAS data set drives the SCL code and defines the class and analysis variables needed for the extraction of the aggregates from the MDDB. The SCL code reads the data set into a SCL list containing sublists. The driver SAS data set contains three variables:

- **Table**: An id to define each aggregate to extract. Also used in naming the output SAS data set
- **Vartype**: Variable type to extract: a class variable (C) or an analysis variable (A)
- **Name**: Name of the variable to extract from the MDDB

File 2 is a PROC PRINT of the driver SAS data set.

![MDDB Extract Driver SAS Data Set - INDSPEC](image)

This data set defines three SAS data sets that will store the defined aggregates.

Figure 3 is the first portion of the SCL code. The SCL code is numbered--match the code to the numbers for the explanations that follow.
Use the Data Set Model and Data Set Data Vector Classes to access the data in the driver SAS data set.

Use this method to assign the driver SAS data set and allow the _GET_ROW_ method to be used to obtain variable values.

A class and analysis variable list are parameters to the MDDB_M Class. Obtain the values of the Table, Vartype and Name variables for each observation in the driver data set and load them into their respective lists.

A master list stores the class and analysis sublists for each level of aggregation to extract. Refer to the POPULATE label in Figure 5 for the SCL code that loads the master list.

At this point in the code, the master SCL list is filled. Figure 4 is a PUTLIST of the Indlist SCL list when all observations from the driver SAS data set are read.

After the driver SAS data set is read into the master SCL list, we are ready to use this for the MDDB extraction. Refer to Figure 5.

1. Load and create an instance of the MDDB_M class
2. Use this method to assign the MDDB to perform the extract of aggregations
3. Obtain the class and analysis list for each level of aggregation to extract. Three temporary SAS data sets will be created.
4. Create a subtable from the MDDB using the class list derived from the driver data set.
Use the analysis list to obtain a subset of the analysis variables from the newly created subtable.

Use the _stats_ list to request only the SUM statistic for the requested analysis variables.

Create a SAS data set name that will store the aggregation.

Write the aggregation to the SAS data set. If more than one analysis variable is requested, transpose the data so it will be stored in row and column format.

At this point, all the data sets containing the aggregations are created. Figure 6 is a PROC PRINT of the first table extract as defined in the driver data set.

![Table](image)

**CONCLUSION**

This paper presented an approach to extract aggregate data from a MDDB in a batch process. In most cases, the OLAP tools for reporting provide a quick and easy way to your data contained in a MDDB. The SAS/AF classes provide the opportunity to add additional functionality and exploit the data contained within a MDDB. Once the data has been extracted to a SAS data set, the power of the entire SAS System is available for additional processing.

**ACKNOWLEDGMENTS**

The author thanks Mark Moorman of SAS Institute for his assistance.

SAS, SAS/MDDB Server, SAS/AF and SAS/EIS are registered trademarks or trademarks of SAS Institute Inc. In the USA and other countries. © indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.
REFERENCES


For more information contact

Michael F. Bretz
ESMPD 1200-4
Bureau of the Census
Washington, D.C. 20233
ccmail: michael_f_bretz@ccmail.census.gov