Paper 150-27

Migrating a Publicly Available Web Database to a SAS® Solution: Building Multidimensional Databases (MDDBs)

Teresa L. Grimes, QRC Division of Macro International, Inc., Bethesda, MD
Heidi L. Clark, QRC Division of Macro International, Inc., Bethesda, MD

ABSTRACT

AppDev Studio™ 2.0 software, SAS/EIS® software, and SAS/MDDB® Server software are being used in a Microsoft® Windows NT® operating system environment to redesign WebCASPAR, a publicly available database of higher education statistics. The data is being stored in multidimensional data sets (MDDBs) and the interface is being created using SAS webAF™ software. This paper discusses issues encountered with SAS/EIS software, PROC MDDB, and SAS/MDDB Server when building MDDBs for the migration of WebCASPAR to a SAS Online Analytical Processing (OLAP) solution:

- Data structure considerations before building the MDDBs. How many MDDBs should be built?
- Migrating metadata from Microsoft SQL Server database tables.
- Comparison and contrast of the methods that may be employed to create MDDBs.

The intended audience is beginning to intermediate SAS/EIS software and SAS/MDDB Server software users.

INTRODUCTION

WebCASPAR (http://caspar.nsf.gov) is a publicly available online database of statistical data resources dealing with science and engineering at U.S. academic institutions. WebCASPAR is funded by the National Science Foundation (NSF) and contains data from NSF surveys of colleges and universities as well as data from the National Center for Education Statistics (NCES) Integrated Postsecondary Education System (IPEDS) surveys. Since it's release in May 1997, over 6,000 users have registered and over 75,000 multidimensional tables have been generated. WebCASPAR users are primarily analysts and researchers at academic institutions and government agencies.

The user interface and database for WebCASPAR were developed by QRC Division of Macro International, Inc. (QRC). The user interface is written in WebInterchange™ (WebIC), a combined interpreter and scripting language developed by QRC, which creates dynamically generated Web pages based on the contents of ODBC (Open Database Connectivity) compliant databases. A WebIC template combines WebIC instructions, SQL (Structured Query Language) statements, and HTML (hypertext markup language) code to dynamically generate a page.

WebCASPAR's metadata are stored in a Microsoft® SQL Server 6.5 database. However, the system's multidimensional database engine that stores and retrieves data is a highly compressed b+ tree file system that was originally developed by QRC in the mid-1980s for WebCASPAR' s predecessor. Data are retrieved from this file system through scripts generated by the WebIC templates. The data in this file system originate from an extensive set of SAS programs that summarizes, edits, crosswalks, and validates data obtained from external sources and stores them in SAS datasets.

Although improvements have been made to the multidimensional database engine, it is essentially the same product that has been in use since it was originally developed by QRC in the mid-1980s. The predecessor to WebCASPAR was developed for a MS-DOS® operating system environment and was distributed via File Transfer Protocol (FTP) download. When WebCASPAR's predecessor was first developed, it was a state-of-the-art system in every respect, comparable to the best in commercially available software. Even when development efforts for WebCASPAR began in the mid-1990s, using the database engine of the original system as the back-end to the Web-based system rather than migrating to commercial database software was deemed to be the most cost-effective and efficient means of delivering the data. However, cost-effective commercial solutions for the processing, storage, and delivery of multidimensional data have since evolved. At the same time, the costs associated with maintaining and improving the current WebCASPAR system have risen and the potential increase in efficiency and functionality that may be achieved using the current technology is limited.

Therefore, the NSF and QRC have deemed it desirable to migrate the WebCASPAR interface and database to commercial products. A SAS OLAP solution was chosen. SAS AppDev Studio software, specifically webAF software, is being used for the redesign of the user interface. SAS/ MDDB server and SAS/EIS software are being used for the database. This paper discusses issues encountered with SAS/EIS software, PROC MDDB, and SAS/MDDB Server software when building MDDBs for the migration of WebCASPAR to a SAS OLAP solution.

DATA STRUCTURE CONSIDERATIONS

WebCASPAR contains data from five NSF academic surveys and six NCES IPEDS surveys. Each of these surveys has one or more database files associated with it. The number of database files and their structure in the current system was primarily motivated by the goal to minimize the sparsity of the data. However, in some cases, the number of files and their structure was based upon technical limitations of the software. Therefore, the data structures for each survey need to be examined to determine if the current structure is the most appropriate and efficient for the MDDB.

The current structure of the NSF Survey of Research and Development Expenditures at Universities and Colleges (academic R&D expenditures survey) in WebCASPAR has data organized into three database files. There were several considerations in deciding whether to combine these data into a single MDDB or build a separate MDDB for each of the corresponding data files in the current system:

- How many common dimensions are there?
- How many common members are there for the YEAR dimension?
- How many measures are there in each of the current files?

The three academic R&D expenditures survey database files for the current WebCASPAR system have all dimensions in common. These five dimensions are:

```
FICE='Academic Institution'
INFICE='R&D Expenditures Survey Institution'
FLD='Academic Discipline'
RDDISC='R&D Academic Discipline'
YEAR=’Year’
```

A given value of a dimension is a member. The YEAR dimension has many members common to the three files:

```
File A:  1972-2000
File B:  1981-2000
File C:  1972-1989
```

The files have the following number of measures:

```
File A:  6 measures
File B:  2 measures
File C:  2 measures
```

Because the three current files share the same five dimensions and have relatively few combined number of measures, it was decided to combine them into a single MDDB. However, because missing values will be stored for the File B and File C measures that aren't available for all of the years for which the File A measures are available, a combined MDDB requires more disk space than the sum of the disk space required for individual MDDBs.

Since WebCASPAR users often want to access measures from different files for a single analysis, the additional disk space required for a single MDDB was deemed an acceptable compromise for the additional processing time that would be required to access and retrieve data from multiple MDDBs.

MIGRATING THE METADATA

The descriptions of all of the measures in WebCASPAR as well as descriptions of their values are stored in Microsoft SQL Server 6.5 database tables. A program using SAS macro language was written to read these data from the SQL tables and load them into a SAS format library that could be used to apply formats to the dimensions and measures of the MDDB. The macro uses SAS/ACCESS® to ODBC and the FORMAT procedure in Base SAS.

```
LIBNAME LIBRARY 'L:\CASPAR6\FORMATS';

%MACRO READSQL(SQLTAB,FMTNAME);

/* Read the SQL database table and create a SAS
table that contains the columns needed for the
FORMAT procedure */

PROC SQL;
  CONNECT TO ODBC
  (DSN=WEBCASPAR UID=******* PWD=******);
  CREATE TABLE &SQLTAB AS
  SELECT * FROM CONNECTION TO ODBC
    (SELECT CODE, NAME FROM &SQLTAB);
  DISCONNECT FROM ODBC;
QUIT;

/* Rename columns in SAS table to correspond to
the column names needed for the FORMAT procedure
and create additional columns, FMTNAME and TYPE,
needed for the FORMAT procedure */

DATA &SQLTAB;
  SET &SQLTAB
      RENAME=(CODE=START NAME=LABEL));
  FMTNAME="&FMTNAME";
  TYPE='C';
RUN;

/ Use the SAS table as the input control data
set for the FORMAT procedure to construct the
format */

PROC FORMAT CNTLIN=&SQLTAB LIBRARY=LIBRARY;
RUN;

%MEND READSQL;

/* Macro invocations: READSQL(SQLTAB,FMTNAME) */

/* cross-survey dimensions */

%READSQL(acadinst,fice);     /* institution */
%READSQL(acaddisci,fld);     /* discipline */

/* R&D expenditures survey dimensions */

%READSQL(rdinst,rdinst);     /* institution */
%READSQL(rddisc,rddisc);     /* discipline */
```

HOW TO BUILD THE MDDB

There are four methods that may be used to build MDDBs:

- PROC MDDB
- SAS/EIS
- SAS/MDDB Server classes
- SAS/Warehouse Administrator® software

In deciding which method to use to build MDDBs, several factors need to be considered:

- The knowledge, skills, and training required of the person building the MDDB.
- The ease of use of the method in defining subcubes to be stored in the MDDB and the related ability to populate the MDDB with subcubes in order to optimize response time.
- The system resources required for the method.
- The cost of the licensed SAS software products required for the method.

In addition to these factors, each method has its advantage:

- When you build an MDDB using SAS/EIS software, the MDDB will automatically be registered in the SAS/EIS metabase facility.
- The SAS MDDB procedure requires the least number of licensed SAS software products.
- SAS/MDDB Server software classes provide more flexibility and control to the developer.

Due to project budget limitations, the SAS/Warehouse Administrator method was not considered.

SAS/MDDB Server software classes were also eliminated from consideration, because the SAS programmers who will eventually be creating, maintaining, and updating the MDDBs are not familiar with object-oriented programming concepts, SAS/AF® software, or SAS Component Language (SCL).

For the remaining methods, SAS/EIS and the SAS MDDB procedure, sample MDDBs were created using each before deciding which method to employ for building all of the system's MDDBs.

SAS/EIS was chosen as the initial method for creating MDDBs, because it forces the creation of metadata, which is stored in a metabase. This metadata is used by SAS webAF® to access and display the MDDB data.

The SAS MDDB procedure was chosen as a secondary method for creating and updating MDDBs when the initial MDDB created requires modification. The interface for SAS/EIS was found to be difficult to use in order to make modifications to MDDBs, in particular with regard to adding and deleting subcubes (hierarchy statements in PROC MDDB). After an initial MDDB has been generated using SAS/EIS, the PROC MDDB source code generated by SAS/EIS can be easily edited to modify the MDDB.

For example, a MDDB was specified and created using SAS/EIS. The "Most Paths Covered" method was used to define the subcubes to be constructed with the MDDB. The following PROC

MDDB code was generated for the MDDB specified using SAS/EIS:

```
proc mddb data=IN.RDEXP out=CUBES.RDEXP
label='NSF Academic R&D Expenditures Survey'
;
class FLD /ASCENDING;
class RDDISC /ASCENDING;
class FICE /ASCFORMATTED;
class INFICE /ASCFORMATTED;
class YEAR /ASCENDING;
var TOTRD / MAX N SUM MIN NMISS;
var FEDRD / MAX N SUM MIN NMISS;
var SLEXP / MAX N SUM MIN NMISS;
var OTHEXP / MAX N SUM MIN NMISS;
var OWNEXP / MAX N SUM MIN NMISS;
var INDEXP / MAX N SUM MIN NMISS;
var TOTEQP / MAX N SUM MIN NMISS;
var FEDEQP / MAX N SUM MIN NMISS;
var TOTCAP / MAX N SUM MIN NMISS;
var FEDCAP / MAX N SUM MIN NMISS;
hierarchy FLD RDDISC FICE INFICE YEAR;
hierarchy FLD FICE INFICE YEAR;
hierarchy FICE INFICE YEAR;
hierarchy FLD RDDISC INFICE YEAR;
hierarchy FLD INFICE YEAR;
hierarchy INFICE YEAR;
hierarchy FLD RDDISC FICE YEAR;
hierarchy FLD FICE YEAR;
hierarchy FICE YEAR;
hierarchy FLD RDDISC YEAR;
hierarchy FLD YEAR;
hierarchy YEAR;
hierarchy FLD RDDISC FICE INFICE;
hierarchy FLD FICE INFICE;
hierarchy FICE INFICE;
hierarchy FLD RDDISC INFICE;
hierarchy FLD INFICE;
hierarchy INFICE;
hierarchy FLD RDDISC FICE;
hierarchy FLD FICE;
hierarchy FICE;
hierarchy FLD;
RUN;
```

For these data, it is known that users will not require a sum of the data for all years. Therefore, the subcubes (hierarchy statements) that do not contain the YEAR dimension are not necessary. The PROC MDDB code may be edited to remove these and then run to recreate the MDDB without the unnecessary subcubes. Alternatively, the MDDB may be updated with PROC MDDB by using the REMOVEHIER statement with the IN= and OUT= options. The IN= and OUT= options are used to specify the input MDDB and the output MDDB, respectively. Hierarchies are removed from the MDDB specified on the IN= option using the REMOVEHIER statement, and the resulting MDDB is written to the file specified on the OUT= option.

CONCLUSION

This simple case study has highlighted some issues encountered in the process of building MDDBs for the migration of a publicly available Web database a SAS OLAP solution. First, the structure of the data was examined and a decision was made whether to create a single MDDB or multiple MDDBs for a given the data source. Next SAS/ACCESS to ODBC and the FORMAT procedure in Base SAS were used to migrate the current system's metadata stored in Microsoft SQL Server 6.5 database tables to a SAS format library. Finally, the available methods for creating MDDBs were evaluated. It was decided to use SAS/EIS to create the initial MDDB for a data source and then use the MDDB procedure to make modifications to the MDDB. The processes described in this paper will be used to migrate the remaining WebCASPAR data sources to MDDBs.

REFERENCES

Greg Henderson, *What You Need to Consider When Building and Deploying an OLAP Application*, Paper 37, Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc., 2001.

Duane Ressler and Mary Simmons, *SAS/MDDB Server Software: Expanding Open Access to Your OLAP Data*, Paper 144, Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc., 2000.

SAS Institute Inc., Introduction to SAS/EIS and SAS/MDDB Server Software Course Notes, Cary, NC: SAS Institute Inc., 2000.

SAS Institute Inc., SAS OLAP Server Administrator's Guide, Release 8.1, Cary, NC: SAS Institute Inc., 2000.

Ian J. Sutton, *Convert Your Model T into a Ferrari – Unleash the MDDB*, Paper 144, Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc., 2000.

Peter R. Welbrock, *Strategic Data Warehousing Principles Using SAS Software*, Cary, NC: SAS Institute Inc., 1998.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Teresa L. Grimes
QRC Division of Macro International, Inc.
7351 Wisconsin Ave., Suite 400W
Bethesda, MD 20814
Work Phone: (301) 657-3077 x127
Fax: (301) 961-8595
Email: tgrimes@qrc.com
Web: www.qrc.com

Heidi L. Clark
QRC Division of Macro International, Inc.
7351 Wisconsin Ave., Suite 400W
Bethesda, MD 20814
Work Phone: (301) 657-3077 x153
Fax: (301) 961-8595
Email: hclark@qrc.com
Web: www.qrc.com