

OLE DB for OLAP: SAS® Institute Leading the Way

Duane Ressler, SAS Institute Inc., Cary, NC

Randy Pierce, SAS Institute Inc., Cary NC

Mark Moorman, SAS Institute Inc., Cary NC

ABSTRACT

Open access to SAS® data warehouses means more than just ODBC access to SAS data sets. Today with the growth of OLAP data stores in a SAS data warehouse it is important to offer open access to multidimensional data stores such as those in SAS/MDDB® server and CFO Vision™ software. With the Microsoft API OLE DB for OLAP, SAS introduces openness to its OLAP data stores through the Open OLAP Server. It is now possible to use popular Windows-based OLAP applications like Excel, Cognos (PowerPlay), and Knosys (ProClarity) to access SAS multidimensional data stores. This support also allows customers to build their own client applications using development tools such as Visual Basic, C++, and scripting languages such as VB Script and JAVA Script.

INTRODUCTION

Microsoft has a long history of defining data access protocols including DAO, ODBC, and now OLE DB and ADO. OLE DB is a base level protocol for establishing a client/server connection and moving rectangular data through this connection. The OLE DB architecture allows it to be extended to accommodate many types of data, and OLE DB for OLAP represents the first such extension, made to accommodate OLAP, or multidimensional, data.

Since SAS warehouses are enterprise wide, it is unreasonable to believe that all users of SAS data would be using a single client tool. As new divisions or departments within a company start making use of the data warehouse, it is important to have as many client options as possible available. With OLE DB for OLAP, there exists a single API capable of delivering data to many different client products. SAS Institute began to implement this Microsoft API in order to efficiently and rapidly surface OLAP data from the data warehouse to many different OLAP clients. By using this API, we have also enabled customers to build custom OLAP clients using popular programming languages and development environments.

This paper will describe how our OLE DB for OLAP support, via the Open OLAP Server, can benefit our customers. Some detail on the API itself is included, but for complete information on the specification, refer to the Microsoft OLE DB for OLAP web site.

www.microsoft.com/data/oledb/olap/

THE SPECIFICATION

The OLE DB for OLAP specification defines a set of Component Object Model (COM) objects, as well as the interfaces for those objects. The specification addresses both the server, or provider, as well as the client, or consumer. An OLE DB for OLAP provider implements the defined objects and their interfaces. A consumer uses the objects and interfaces to request data from a provider. Because of the clearly defined interfaces in the OLE DB for OLAP specification, an OLAP consumer can request and receive data from any provider implementation. Likewise, an OLAP provider can supply data to any consumer. Because SAS Institute has chosen to implement an OLE DB for OLAP provider, any consumer that complies with the OLE DB for OLAP specification can consume OLAP data from the SAS system.

The OLE DB for OLAP specification is an extension of Microsoft's OLE DB specification. The OLE DB specification describes how a consumer contacts a data source, acquires metadata about the

data source, and sends commands to the data source to retrieve specific data. With an OLE DB data source, these commands are typically SQL statements and the retrieved data is usually a table with rows and columns. Because the OLE DB specification can be easily extended to accommodate other types of data, Microsoft defined the OLE DB for OLAP extension to handle multidimensional data.

The OLE DB for OLAP specification describes a new command language called Multidimensional Expressions (MDX). Instead of returning data in a tabular form, these commands return the data in a multidimensional form. To extend the OLE DB specification to accommodate OLAP data, one additional COM object was added to the specification with two new interfaces. The Dataset object represents data returned from execution of an MDX command. Consumers use the new IMDDataset and IMDFind interfaces to navigate the returned data and present it to the user.

SAS Institute has been closely involved in the development of the OLE DB for OLAP specification and made every effort to ensure the specification will support the needs and requirements for OLAP data that are most important to our customers. These efforts include providing important feedback to Microsoft on technical aspects of the specification, as well as working with other software vendors on proof of concept efforts on the OLE DB for OLAP specification. In fact, at Microsoft's invitation, SAS was the first company to demonstrate a working implementation of OLE DB for OLAP and successfully connected to multiple early consumer implementations.

SPECIFICATION PRIMER

In order to begin to understand the concepts behind the OLE DB for OLAP specification, it is important to look at the following topics in more detail:

- Multidimensional schemas
- Multidimensional expressions (MDX)
- Datasets

MULTIDIMENSIONAL SCHEMAS

OLE DB for OLAP provides interfaces for multidimensional data providers (MDPs) to expose a set of multidimensional cubes, or schemas, and for multidimensional consumers to view these schemas. OLE DB for OLAP also provides interfaces that allow users to browse a set of metadata objects for information about multidimensional data.

The primary metadata object defined by OLE DB for OLAP is the cube. A cube is a set of related dimensions, similar to the hierarchies or drillpaths in SAS/EIS® software. These dimensions define a multidimensional data space. Each data point in this data space is uniquely identified by a set of coordinates. Each coordinate is a member of one component dimension.

A cube is made up of a set of dimensions and each dimension consists of a set of members. Just as a dimension is similar in concept to a hierarchy or drillpath in SAS/EIS software, the members of a dimension are the union of the values of the classification variables in that drillpath. In the OLE DB for OLAP definition, the members of a dimension can be aggregated along a hierarchy, with some dimensions having more than one hierarchy. Each hierarchy has levels, equivalent to classification

variables in the SAS multidimensional model.

A multidimensional consumer can examine several pieces of metadata from an MDP, including:

- cubes
- schemas, a collection of cubes
- catalogs, a collection of schemas, with no relationship to a SAS catalog
- dimensions
- measures, a special type of dimension
- hierarchies
- levels
- properties (for each dimension level)
- members

For example, consider the ProdSale sample MDDB supplied with SAS EIS software. Examining it from an OLE DB for OLAP perspective, this cube includes four dimensions: Geographic, ProductLine, Time, and the special Measures dimension. Each of these dimensions contains a single hierarchy consisting of all the levels for the dimension. The Measures dimension contains a single level, whose members consist of a combination of the analysis/statistic pairs in the MDDB. The Geographic dimension contains three levels: Country, Region, and Division. The ProductLine dimension contains two levels: ProductType and Product. The Time dimension contains three levels: Year, Quarter, and Month.

MULTIDIMENSIONAL EXPRESSIONS (MDX)

Once a multidimensional data consumer connects to a data provider, the consumer can submit a query to request data from the multidimensional provider. A query may request data from one or more cubes and extracts the data into a single, multidimensional cube. Typically, the cubes requested by a query contain no more than 4 dimensions and are often no more than 2 or 3, depending on the display capabilities of the data consumer on the client. These queries are similar to relational SQL queries that request data from one or more tables and return the results in a single table.

An MDX query must include:

- One or more cubes that set the scope of the query
- The number of axes
- The dimension(s) projected on each axis and level of nesting at which each dimension appears on the axis.
- The members from each dimension to include on the axis and their sort order
- The members included on the filter axis.

An MDX statement takes the form of:

```
SELECT <axis specification> [, <axis specification> ...]
FROM <cube specification>
WHERE <filter specification>
```

When making a query, the user or application divides the dimensions into two groups: axis dimensions, which specify the dimensionality of the cube and filter dimensions for which data is retrieved for a single member. For example, the following query specifies members of a single dimension on each of two axes and a single member of two dimensions on the filter axis.

```
SELECT {[1993], [1994]} ON COLUMNS,
{[CANADA], [EAST_CANADA], [U.S.A.],
[WEST_U.S.A.]} ON ROWS
```

```
FROM [SAMPLE]
WHERE ([PRODUCT LINE:-ALL-],[PREDICT_SUM])
```

Notice that this query specifies each dimension: Time on the columns, Geographic on the rows, and ProductLine and Measures on the filter, or subset, axis.

There is a definite similarity between MDX and SQL. In fact, the OLE DB for OLAP specification includes information on mapping MDX queries into SQL queries to allow multidimensional access to relational data.

While MDX in itself is a very powerful query language, the true power of MDX comes through the use of client applications that submit MDX to a provider. These client applications serve to insulate the user from the MDX language, allowing them to query data using a point and click environment.

DATASETS

Execution of an MDX query returns a dataset. Unlike SAS data sets, which store tabular data, an OLE DB for OLAP dataset object returns multidimensional data. A dataset consists of a set of axes, with the points along each axis being called coordinates. If an axis consists of a single dimension, then the coordinates are a subset of the members of the dimension. If an axis contains more than one dimension, then each coordinate consists of a member from each of the dimensions displayed on the axis.

So, looking at the ProdSale query shown above, the returned dataset would look like this (represented in a simple table form):

	1993	1994
Canada	\$119,329.00	\$113,690.00
East	\$62,426.00	\$58,220.00
U.S.A.	\$123,763.00	\$117,959.00
West	\$60,484.00	\$60,651.00

OLE DB FOR OLAP AND ACTIVEX DATA OBJECTS

Just as ActiveX Data Objects (ADO) are built on top of the OLE DB specification, Microsoft has built ADO MD (multidimensional) on top of OLE DB for OLAP. ADO MD allows application and web site developers to easily access data from the SAS OLE DB for OLAP provider using Visual Basic, JavaScript, or Java.

THE SAS OLE DB FOR OLAP PROVIDER

Like most providers, the SAS multidimensional data provider, Open OLAP Server, will expose a subset of OLE DB for OLAP metadata objects and MDX capabilities. The OLE DB for OLAP specification is intended primarily to surface data to Microsoft specific client platforms, however, the specification allows for providers to connect to multidimensional servers in any platform or architecture. The SAS OLE DB for OLAP provider, running on Windows 95, Windows NT Workstation, and Windows NT Server, is designed to connect to any platform running release 6.12 of the SAS System where SAS/EIS Software and SAS/MDDB Server Software products are licensed. This portable connection will allow you to easily convert existing applications into a thin client environment.

The Open OLAP Server consists of both client and server components. The client components include:

- The provider, which allows your Windows 95 or Windows NT PC to serve up data from any Open OLAP Server to any OLE DB for OLAP compliant consumer, and

- The server administrator, which allows you to remotely administer an Open OLAP Server from your PC.

The server component includes SAS catalogs, which can be used with any SAS 6.12 TS045 or higher installation of the SAS system to turn it into an Open OLAP Server.

The Open OLAP Server supports the following consumers:

- Cognos (PowerPlay)
- Knosys (ProClarity)
- Microsoft Corporation (Excel 2000)
- Seagate Software (Crystal Reports)
- Seagate Software (Worksheet)

REGISTERING YOUR DATA

In order for you to use the OLAP data from your data warehouse with the SAS Open OLAP Server, it is necessary to register additional metadata for your data. Because the OLE DB for OLAP specification defines the presentation of OLAP data in a more constrained manner than does SAS/EIS software, you will need to register additional information about dimensions, hierarchies, levels, and measures for your OLAP cubes. This additional registration can be performed using a metadata registration interface provided with the Open OLAP Server.

When you register an OLE DB for OLAP dimension, which is similar to a drillpath from SAS/EIS software, you can define:

- Dimension levels, which are the SAS variables that constitute a dimension,
- The dimension type,
- The default member for the dimension
- The dimension description, and
- The dimension caption.

For each dimension level, you can register metadata on:

- The level caption, a label associated with the dimension for display purposes,
- The level description, and
- The level type.

In addition, for the Measures dimension, you can select the measures that you want to be available to consumer applications.

The Open OLAP server documentation provides additional information on registering this metadata and describes how to keep the metadata in sync with the data cube as it is updated.

Users of CFO Vision software do not need to register additional metadata in order to surface data from the CFO Vision OLAP store via the Open OLAP Server.

SURFACING YOUR DATA

The Open OLAP Server is designed to surface OLAP data from a running SAS session. This server session is typically run on a server class machine and the OLAP data to be served should be directly accessible from this machine. For example, you could run SAS on a UNIX box and surface Hybrid OLAP or SAS/MDDDB Server data from that box, through the provider to client applications running on a Windows 95 or Windows NT machine.. Alternatively, you could run the server on a Windows NT machine and surface the data to client applications on the same or other Windows NT machines.

This SAS server session must be running release 6.12 of the SAS System (TS045 or higher) and have both a SAS/MDDDB Server software and SAS/EIS software license. Any client

machine with TCP/IP access to this server can query the data surfaced from this SAS session via the provider.

USING CLIENT APPLICATIONS

Customers who are interested in connecting an existing OLE DB for OLAP compliant consumer to the Open OLAP Server to view data maintained in a SAS Data Warehouse or in CFO Vision software can easily do so. Until there was a single widely adopted API for OLAP data, there was no universal way to surface OLAP data from the SAS system to a wide range of OLAP presentation tools. Because the Open OLAP Server implements the OLE DB for OLAP specification, our data is now available to any specification compliant consumer.

The provider portion of the SAS Open OLAP Server is a small .DLL that resides on the client machine. This .DLL is registered with the Windows operating system as an OLE DB for OLAP provider. This registration allows consumer applications to detect providers on the user machine. The user begins the connection to the Open OLAP Server by selecting it as the provider from the consumer application. After that selection is made, it is necessary to a running SAS Open OLAP Server either by its node name or actual numeric TCP/IP address. Users of CFO Vision software will need to enter their CFO Vision userid and password. Users of SAS/MDDDB Server software may need to know the password for the server, if that server was started in a password protected mode. Once the server has been identified and any required passwords are verified, the consumer simply uses objects and interfaces defined in the specification to display the available data and then query that data as directed by the user.

When necessary, consumer calls to the provider are forwarded to the running SAS Open OLAP Server. The server will extract the requested data and return it to the provider. The provider, in turn, returns the data to the consumer as dictated by the OLE DB for OLAP specification. Multiple users can contact the same server simultaneously. And each user can be using a different consumer. The server simply knows that a provider wants data, and returns it. The server does not keep a state for each consumer connected to the provider.

BUILDING A CUSTOM CONSUMER VIA COM

Before the development of the Open OLAP Server, SAS customers that wanted a custom build OLAP tool had little choice but to use a SAS language to implement that application. This approach could require a significant development effort using SAS/EIS software and SAS/AF® software. However, since SAS now has an OLE DB for OLAP provider, any development language that supports COM can be used to contact the provider and thus access any multidimensional data being served by an Open OLAP Server.

To learn more about the actual COM objects and interfaces, refer to the documentation and materials in the Microsoft Data Access SDK. The SDK includes the OLE DB for OLAP specification and help files. Although any COM language could be used to drive the objects, the SDK only includes sample C++ code. Further, Microsoft Visual C++ release 6.0 includes a number OLE DB for OLAP related technologies. There are numerous ATL (Active Template Library) templates to ease implementation of a consumer, and even a wizard that generates the basic structure of an OLE DB consumer for you.

BUILDING A CUSTOM CONSUMER WITHOUT USING COM

For users who want the flexibility of building their own customer consumers but do not want to learn the complexities of COM, Microsoft has provided a set of scriptable COM objects (ActiveX objects) that make calls to an OLE DB for OLAP provider. From a provider standpoint, these objects act like any other consumer. They contact a provider, ask for metadata, execute MDX commands, and then use the IMDDataset and IMDFind interfaces

to navigate the returned data.

However, these objects do not present the data as do most consumers. They make calls to the provider only when directed to do so programmatically. These objects can be driven from VB, VBA, VB Script, JAVAScript, and other object based languages. They can even be used by C++ programmers who do not want to make direct COM calls to the provider. The scriptable objects that overlay OLE DB itself are called ADO (ActiveX Data Objects). ADO was extended with ADO MD to provide scriptable objects that overlay OLE DB for OLAP objects. The Microsoft Data Access SDK comes with ADO, ADO MD, and documentation for both sets of objects.

CONCLUSION

Using the OLE DB for OLAP API as the basis of the Open OLAP Server, SAS Institute is offering the most open solution to its customers. It is possible using the Open OLAP Server to access SAS multidimensional data stores via compliant clients with no additional programming. Offering customers the ability to use tools as diverse as SAS/EIS software and MS Excel with the same multidimensional data store creates a true enterprise data warehouse. By extending that multidimensional data store beyond a Windows only server, SAS makes it possible to have all the options open when building OLAP applications. Our customers will not be limited by client, server, data store, or programming tools.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the authors at:

Duane Ressler
SAS Institute Inc.
SAS Campus Drive
CARY NC 27513
Work Phone: 919.677.8000 x6950
Email: sasdyr@wnt.sas.com

Randy Pierce
SAS Institute Inc.
SAS Campus Drive
CARY NC 27513
Work Phone: 919.677.8000 x7240
Email: sasjwp@wnt.sas.com

Mark Moorman
SAS Institute Inc.
SAS Campus Drive
CARY NC 27513
Work Phone: 919.677.8000 x6522
Email: sasmwm@wnt.sas.com

SAS, CFO Vision, SAS/AF, SAS/EIS, and SAS/MDDB 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.