



MDX Myths, Facts and the Future

By Aryn Rajan
Simba Technologies Inc.

I'm often asked why companies would want to build an MDX Provider – either OLE DB for OLAP (ODBO) or XML for Analysis (XMLA). It is an interesting question in that the companies asking generally have an analytical reporting solution that can be abstractly broken into two parts – a multi-dimensional or OLAP data source and a query and reporting interface on top of the data source.

The company typically has a great analytical reporting solution that their customers are happy with. The question often starts with one of the company's existing customers wondering if they can use other analytical or OLAP reporting tools – usually something as simple as Excel Pivot Tables – with the company's product. Other times, the question starts with a prospective customer saying that they want to purchase the analytical reporting solution, but they want to use other front ends – again, usually Excel Pivot Tables.

Invariably, someone in the company does some quick research

and realizes that they need to support the MDX language and the ODBO interface in order to have Excel Pivot Tables connect to their OLAP data source. They quickly realize that MDX is a complex language, and it would be a lot of work to build MDX into their OLAP data source. So, the person looks for a solution to simplify the process of building an MDX engine, and of course, the third-party solution they find is Simba.

Building an ODBO or XMLA Provider complete with an MDX engine is no easy feat. If you want to open your multi-dimensional data source using the standard MDX language via the ODBO or XMLA interfaces, your options are to either build it yourself, use SimbaProvider SDK, or move the data into another product like Microsoft Analysis Services, SAP Business Warehouse, or Hyperion Essbase and serve the data from there.

Below is a simple table showing the pros and cons of these three options:

	Build Your Own	Use Microsoft Analysis Services, SAP BW or Hyperion Essbase	Use SimbaProvider SDK
Cost	Very high	Your customers need to license one of these products every time they want to buy from you	Reasonable
Time	Months to Years	As long as it takes to move the data	Weeks to Months
Knowledge Requirement	Your developers need to learn MDX and know how to build an efficient MDX Engine	Minimal	Minimal
Maintenance	You must keep a team of MDX experts to support your product	Minimal	Minimal
Compatibility	You need to make sure you test with all new versions of all MDX reporting tools	Microsoft, SAP, or Hyperion deals with this	Simba deals with this
Location of Data	In your multi-dimensional or OLAP data store	In Microsoft's, SAP's, or Hyperion's products	In your multi-dimensional or OLAP data store



Given the complexity of the technology and the limitations and ongoing costs of using a product such as Microsoft's, SAP's, or Hyperion's, most companies opt to add MDX and ODBO or XMLA interfaces with Simba's Software Development Toolkit (SDK).

Simba's OLAP product is called SimbaProvider SDK. SimbaProvider contains a very powerful MDX Engine that can be put on top of pretty much any multi-dimensional or OLAP data source. The SDK accepts an MDX query through either the ODBO or XMLA interfaces, parses the MDX query, and then via the MDX Engine, breaks the MDX into multi-dimensional components that are mapped to the multi-dimensional API of the underlying multi-dimensional or OLAP data source. If a proprietary multi-dimensional or OLAP data source supports an API that allows access to dimensions, hierarchies, members, etc., then SimbaProvider SDK can be used to quickly develop an MDX/ODBO/XMLA interface on top of the data source with a minimal amount of work.

If you had to build your own MDX interface on top of your multi-dimensional or OLAP data source, it would probably take months to years to complete the work. By using an

SDK like Simba's, the effort drops dramatically, allowing you to add an MDX interface in a fraction of the time. To fully understand MDX's enormity, you'd benefit to ask MDX experts like George Chow, Bruce Johnston, Mosha Pasumansky, or George Spofford about how challenging it is to build a good MDX Engine.

Another question often asked is whether a company should implement ODBO or XMLA. My answer is that ODBO versus XMLA is the easy question. The hard work is building the MDX engine. However, with SimbaProvider, you get an MDX Engine and both the ODBO and XMLA interfaces in one package. You don't need to be an expert in COM (Microsoft's Component Object Model), which forms the basis of ODBO, and you don't need to be an expert in .NET and Web Services, which is the basis for XMLA. While most MDX/OLAP query and reporting products today primarily support ODBO, this is changing. More and more products are supporting XMLA. Also, with the release of Microsoft Analysis Services 2005, my guess is that the push to XMLA will increase substantially.

Here are two tables listing a number of companies that support ODBO and/or XMLA:

ODBO/XMLA Providers

Company	Website
Aleri	www.aleri.com
ALG Software	www.algsoftware.com
Applix	www.applix.com
Descisys TeraSolve	www.descisys.com
Hyperion Essbase	www.hyperion.com or www.essbase.com
INEA	www.ineacorp.com
Microsoft Analysis Services	www.microsoft.com
MIS AG Alea	www.misag.com
SAP BW (NetWeaver)	www.sap.com
SAS	www.sas.com
SunGard (Whitelight)	www.sungard.com/products_and_services/stars/whitelight.htm



ODBO Consumers

Application	Company
Analyst, Smart Client, Open Client	DataBeacon
AOS (Active OLAP Suite)	Application Consulting Group Inc.
BusinessObjects	Business Objects
Business Scorecard Manager	Microsoft
Crystal Analysis	Business Objects
Data Analyzer	Microsoft
DecisionWare	MIS AG
dynaSight	ArcPlan
Excel	Microsoft
Executive Viewer	Temtec
Intelligence	Hyperion (Brio)
Office Web Components	Microsoft
PowerPlay	Cognos
ProClarity	Microsoft (ProClarity)
ReportNet	Cognos
Reporting Services	Microsoft
XLCubed	XLCubed

So what's the future of MDX? The future of MDX is very bright. The reality is that MDX is supported by a large number of products and the list continues to grow. MDX is the primary interface to products like Microsoft Analysis Services and SAP Business Warehouse, and it would take some major effort to change that. Additionally, the XMLA Council, the organization responsible for advancing the XMLA standard, has worked to promote adoption. The XMLA Council was quite active from 2001 until early 2004. Even though not much has happened with the XMLA Council recently, things appear to have standardized pretty well, and the various companies involved are spending more time focusing on actually building and enhancing their products these days.

When trying to understand the MDX language, let me point out some basic facts. There has never been a standalone MDX specification. The MDX specification was always part of the OLE DB for OLAP (ODBO) specification, which was last published in 1999. When the XMLA specification was

published, it referred to the ODBO specification for details on MDX. The last version of the XMLA specification (version 1.1) was published in 2002. So, now you know where all the specifications are. This is nice to know, but you also need to understand that there is no reference implementation of the MDX language. As such, Microsoft Analysis Services has become the de facto reference implementation to which all MDX Engines are compared. Also, with Analysis Services 2005, Microsoft has made numerous extensions to the MDX language. If you are trying to develop an MDX Engine, you need to know what these extensions are, or your implementation will look weak relative to Microsoft. Simba has spent considerable effort in understanding the Microsoft extensions to the MDX language and it has incorporated many of these MDX language extensions into its SimbaProvider SDK.

The future of MDX and XMLA is surely growth. Some published studies have identified Microsoft Excel as the most



popular BI product in the marketplace. In my experience, I would tend to agree with this finding. Excel seems to have a life of its own. If you look at Excel, its Pivot Tables component connects to an MDX data source through the ODBO interface. Excel Pivot Tables connectivity is a must for any data source that supports MDX.

Notably, there are three flavors of Microsoft Excel – Excel 2000, Excel 2003/XP, and Excel 2007 – and the MDX support required for each of these is different to the point that I would consider them three different consumers when it comes to MDX. The most interesting Excel flavor is of course, Excel 2007. Microsoft has made some very major enhancements to the BI functionality in Excel 2007, and if you are building an MDX Engine, you must understand these enhancements. In fact, if you compare Excel 2007, running against Microsoft Analysis Services 2005, versus Excel 2007, running against Microsoft Analysis Services 2000, you will notice how much of the new functionality in Analysis Services 2005 Excel 2007 actually does use. Therefore, if you need to have Excel connect to your multi-dimensional/OLAP data source, the amount of the MDX language that you need to support will increase accordingly. The vast majority of enterprise users have Excel on their desktop, and the only way Excel can connect to a multi-dimensional/OLAP data source is via MDX. Consequently, this is a very good business case for ensuring that you have MDX support in your multi-dimensional/OLAP data source.

Another thing that I see driving the need for MDX support in multi-dimensional/OLAP data sources is Microsoft's new set of BI products within the Microsoft Office family. The first of these is Business Scorecard Manager (BSM). This product sells separately from Office for about US\$5000 per server. I anticipate that there will be more products like BSM from recent Microsoft press releases and from seeing what is happening in the industry. It looks like Microsoft is going to make a big push into the BI space from a reporting/front-end point of view. With Microsoft's announcement that it will acquire ProClarity, I think this is a clear indication that it is extremely serious about being a major player in the BI space. ProClarity's use of MDX, when connecting to Microsoft Analysis Services, is considered very advanced, and this connectivity will only improve.

Of course, I would be remiss in not mentioning Microsoft's Reporting Services, which ships in the SQL Server package. Version 2 of Reporting Services just shipped as part of SQL Server 2005, and Reporting Services also requires MDX support to connect to multi-dimensional/OLAP data sources. You might now see a pattern emerging. Again, if you need a business case to add MDX support into your multi-dimensional/OLAP data source, watch where BSM, ProClarity and other BI products from Microsoft are going.

With respect to SAP BW, people often ask me about SAP's BAPI. MDX is the primary language for query and reporting against SAP BW, and regardless of whether you connect to BW via BAPI, ODBO, or XMLA, you still come to the same MDX engine. Because of this, essentially there are no major differences between the different interfaces to BW. The only difference between the interfaces to BW stem from the communication protocol used – BAPI, ODBO, or XMLA. I have heard and read much about how one query and reporting product is much better than another because of the interface it uses to connect to BW. In my opinion, this is marketing bunk because at the end of the day, everything goes to the same MDX engine, and in reality, if you are doing something that is very query engine intensive, the MDX-based query engine within BW is what determines performance rather than the protocol used to get to the query engine. Having said that, BAPI and ODBO are comparable in performance, whereas XMLA – being a web services interface – will be slower.

Hyperion's Essbase currently supports MDX, but only via XMLA. At Hyperion's 2003 Solutions Conference, Hyperion demonstrated an ODBO Provider for Essbase that was based on Simba's SimbaProvider SDK. In fact, it showed Microsoft Excel connectivity directly from Pivot Tables to Essbase. Hyperion never released an ODBO Provider for Essbase, but considering that an MDX engine already exists along with an XMLA Provider, an ODBO Provider for Essbase would be relatively simple to build. If you are interested in an ODBO Provider for Hyperion Essbase, I would suggest you contact Simba and get an evaluation version of SimbaO2X, which translates MDX queries from ODBO to XMLA.



About the Author

Amyr Rajan is President and CEO of Simba Technologies Inc. Amyr has over 15 years experience in custom software development, and is responsible for driving Simba's success as a leader in data connectivity solutions.

Simba Technologies Inc. is the recognized world leader in standards-based data access products and solutions. Simba works with the world's leading software companies to deliver first class data connectivity solutions.

Simba is a pioneer in ODBC, MDX, ODBO and XMLA. Since 1991, Simba has developed advanced data access solutions for thousands of end users. Today, more than half of all MDX providers have been built with Simba technology, and through a partnership with Microsoft, Simba's SQL technology has been installed on more than 30 million desktops worldwide.

©2007 Simba Technologies Inc. All Rights Reserved.
Printed in Canada.

Simba Technologies Incorporated

938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5

Tel. 604.633.0008

Fax. 604.633.0004

Email. solutions@simba.com

www.simba.com

Simba and the Simba logo are trademarks of Simba Technologies Inc. All other trademarks or service marks are the property of their respective owners. Printed in Canada.