The Intelligencia Spreadsheet Interface

&

MDXL Formula Language

MASS ADOPTION OF BUSINESS INTELLIGENCE

A major objective at iT-Workplace is making the full power of multi-dimensional data accessible to business users. Microsoft has made it clear that they expect these users to access corporate data warehouse using the Microsoft Office suite. The argument for delivering multi-dimensional reporting via Office is a powerful one and could lead to an explosion of both users and multi-dimensional solutions based on its SQL Server Analysis Services database. So far, however the explosion of BI users has not materialized.

The tool that Microsoft has invested in to deliver access to cubes is Excel. In Office 2007 Excel pivot table services has been enhanced to provide better access to multi-dimensional cubes and this functionality is integrated with several other enhancements that allow the creation of attractive dynamic data exploration applications and dashboards. Whilst Excel pivot tables have undoubtedly improved, there are still many limitations that make it difficult and in some cases impossible, to build typical business reports from the warehouse.

One major omission from pivot tables is the lack of ability to work with query based calculations. This forces users to implement all report logic in Excel as cell formulae which can lead to some significant issues:

- A single query calculation can generate hundreds or thousands of consistently applied results in a table. In Excel each of these results would need to be implemented as an individual cell formula leading to significant effort expended and the possibility of formula errors.
- Adding calculated rows and columns to pivot tables is often not possible leading to compromised report design through the use of data sheets or individual cell queries to the OLAP database. This increases complexity and compromises report maintenance.

The problem with providing access to query based calculations is that they need to be defined using the MDX query language that is also used to extract data from multi-dimensional cubes. MDX is a very powerful query language but has a fearsome reputation, even within the database community, for being verbose and difficult to understand. Two factors contribute to the difficulty in understanding MDX:

1. Most people find it difficult to think in more than 2 dimensions making it difficult to visualize how a calculation might be resolved in a complex query. It is common to have 4 or more dimensions displayed in a multi-dimensional query, for example Customer, Product, Salesperson and Time. A single query calculation may interact with all these dimensions at the same time to generate results.

2. The multi-dimensional expressions that define the data in a cube are very long and therefore not “human readable”. For example the field description for an organization might be:

   ```
   [Organization].[Organizations].[North American Operations].[USA Operations].[Central Division]
   ```

   Whilst this expression is actually quite self-explanatory even simple calculations can become extremely long and difficult to read.

COMPARISON TO SQL

It is interesting to compare the adoption of relational/SQL based solutions to the world of multi-dimensional databases. Whilst SQL would hardly be described as a simple language the 2 key problems described for cubes do not exist for relational data sources. Columns typically have short, simple names and the shape of a table is strictly 2 dimensional. The barrier to adoption is therefore reduced as a business user is able to visualize the problem space when defining queries and calculations.

THE SPREADSHEET INTERFACE

Building a basic multi-dimensional query is actually very easy and intuitive for a business user. One of the great powers of a multi-dimensional cube is that the structure of the data, as described by dimensions and hierarchies, is very easy to understand and can be navigated through a graphical user interface without the need to write query expressions. There are many multi-dimensional tools, Excel included, that allow users to build a query by selecting and dragging fields onto a rectangular grid. These tools all generate MDX queries "behind the scenes" but the user is never exposed to the actual query syntax.

A typical multi-dimensional query created in Excel

Amount	Column Labels					
	CY 2001		CY 2002		CY 2003	CY 2004
Row Labels	Actual	Budget	Actual	Budget	Actual	Actual
Corporate	$1,874,469.00	$3,000,480.00	$4,511,243.00	$2,583,420.00	$4,709,851.00	$1,513,940.00
Executive General and Administration	($69,091.00)	($65,940.00)	($147,800.00)	($79,860.00)	($113,826.00)	($65,985.00)
Inventory Management	($193,981.00)	($186,510.00)	($450,696.00)	($226,290.00)	($441,469.00)	($229,248.00)
Manufacturing	($135,012.00)	($134,070.00)	($274,838.00)	($145,770.00)	($277,494.00)	($139,304.00)
Quality Assurance	($79,685.00)	($78,300.00)	($186,619.00)	($82,590.00)	($199,389.00)	($108,873.00)
Research and Development	$3,684,062.00	$4,739,160.00	$9,362,715.00	$4,533,690.00	$10,779,651.00	$4,798,716.00
Sales and Marketing	($53,447.00)	($53,430.00)	($183,002.00)	($49,830.00)	($269,837.00)	($133,555.00)

When defining calculations however the process becomes more difficult and we need to address the 2 problems outlined at the beginning of this document:

1. The interface must help the user to visualize the multi-dimensional workspace in a way that they can understand.
2. A terse, human readable, way to create multi-dimensional expressions is needed.

Intelligencia addresses the first problem by presenting the grid with its own set of row and column labels. This provides a familiar spreadsheet analogy which can be used to address both the numeric data cells and the cube fields that comprise a query result set.

	A	B	C	D	E	F	G
1		CY 2001		CY 2002		CY 2003	CY 2004
2		Actual	Budget	Actual	Budget	Actual	Actual
3	Executive General and Administration	-69,091	-65,940	-147,800	-79,860	-113,826	-65,985
4	Inventory Management	-193,981	-186,510	-450,696	-226,290	-441,469	-229,248
5	Manufacturing	-135,012	-134,070	-274,838	-145,770	-277,494	-139,304
6	Quality Assurance	-79,685	-78,300	-186,619	-82,590	-199,389	-108,873
7	Research and Development	3,684,062	4,739,160	9,362,715	4,533,690	10,779,651	4,798,716
8	Sales and Marketing	-53,447	-53,430	-183,002	-49,830	-269,837	-133,555
9	Corporate	1,874,469	3,000,480	4,511,243	2,583,420	4,709,851	1,513,940

The second problem is addressed in Intelligencia via a familiar looking formula language called MDXL that can be used to create calculations by addressing cube data using spreadsheet references.

THE MDXL FORMULA LANGUAGE

MDXL is designed to bridge the conceptual gap between MDX and Excel. The objective is to allow users that are comfortable building formulae in Excel spreadsheets to build calculations as part of multi-dimensional queries without needing to significantly change their skill level. The basic syntax of MDXL maps multi-dimensional cube data into row and column references allowing n dimensional queries to be visualized as 2 dimensional workspaces. The key concepts of MDXL are:

1. The syntax for building formulae is based on, and extends, the syntax used by Excel. Cube data can be addressed using a row and column reference from the 2 dimensional grid used to display the query

result set. Extensions to Excel syntax are provided to cope with key differences that occur because the underlying data structures are multi-dimensional.

2. MDXL dynamically maps cube data to the result set. If the shape of the query is changed, for example by expanding the detail within a hierarchy, the MDXL expressions will be re-generated so that they correctly address the new result set. [1]
3. MDXL allows the definition of a single formula that applies to many cells in a cube. The user should not normally need to copy cell formulae to multiple cells in the result set.

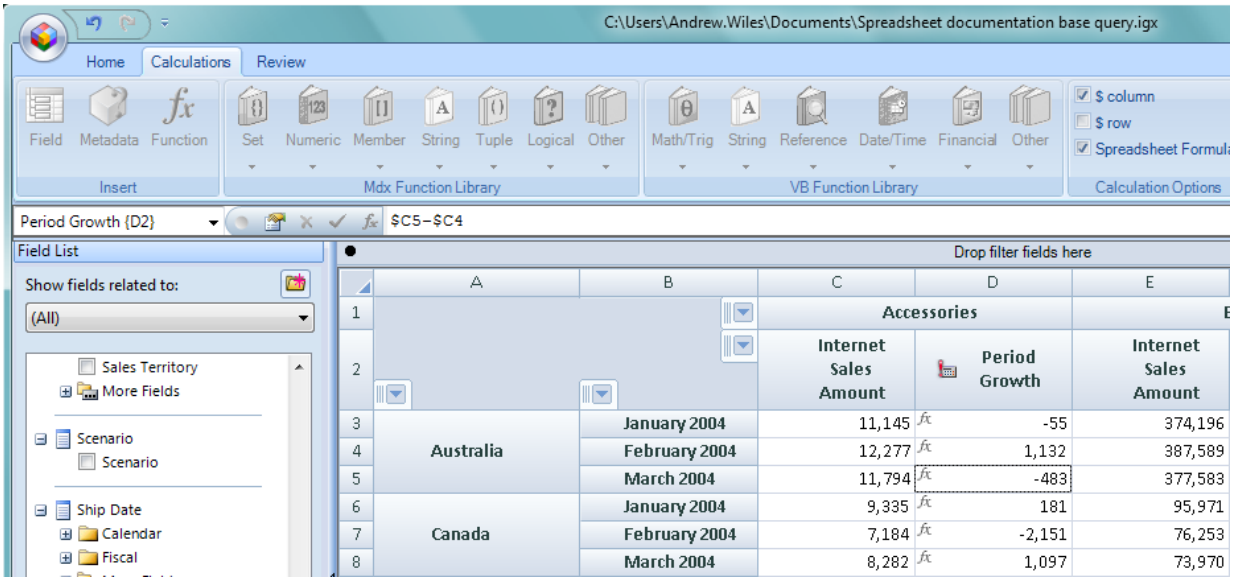
A WORKED EXAMPLE

The best way to illustrate the benefit of MDXL is through an example. For this example we will create a period growth calculation showing how much the Internet sales has increased (or decreased) when compared to the previous period in time. The illustration below shows a query built with 4 hierarchies representing product categories, measures, customer geography and time. The query contains a calculated column named "Period Growth" which has not yet been defined.

	A	B	C	D	E	F	G	H
1			Accessories		Bikes		Clothing	
2			Internet Sales Amount	Period Growth	Internet Sales Amount	Period Growth	Internet Sales Amount	Period Growth
3	Australia	January 2004	11,145		374,196		6,264	
4		February 2004	12,277		387,589		5,458	
5		March 2004	11,794		377,583		7,065	
6	Canada	January 2004	9,335		95,971		5,145	
7		February 2004	7,184		76,253		4,679	
8		March 2004	8,282		73,970		4,325	
9	France	January 2004	4,594		99,013		2,312	
10		February 2004	5,782		156,280		2,292	
11		March 2004	6,186		120,119		1,584	
12	Germany	January 2004	5,472		146,924		2,126	
13		February 2004	5,372		134,556		1,913	
14		March 2004	4,475		162,319		2,255	
15	United Kingdom	January 2004	5,768		149,892		3,067	
16		February 2004	5,895		178,709		2,239	
17		March 2004	6,796		194,087		2,761	

[1] Note: The dynamic mapping used in MDXL is the subject of US patent application 11/751,919

To create the formula in MDX the user would first select the cell D5 to indicate this is the target for the calculation and would then enter the formula $C5-$C4 either by typing or by clicking in the relevant cells in the grid. When the formula is resolved the result will be as shown below.



You will see that the calculation applies to all the cells in each instance of the calculated column and displays the correct values for each sub-group of data ("Accessories", "Australia", "Canada" etc.). The MDX expression generated by the MDXL is:

```
([Date].[Calendar].CurrentMember, [Measures].[Internet Sales Amount])-
([Date].[Calendar].CurrentMember.PrevMember, [Measures].[Internet Sales
Amount])
```

The importance of MDXL can be seen in the brevity and clarity of the formula when compared to the MDX version.

CELL ADDRESSING

Cell addressing in MDXL is based on the syntax used in Excel to determine formula behavior when copying cells. Excel uses the $ symbol combined with row and column references to determine whether a formula relates to an "absolute" or "relative" reference when copying. The expression $B3 indicates that when copied the column reference should be absolute, i.e. always column B and the row referenced should be indexed relative to the original cell from which the formula is being copied. The following table lists the shorthand notations for working with relative and absolute cell addresses.

Notation	Example	Description
$<col>$<row>	C4	Absolute reference to a specific cell or group of cells that are defined by the innermost labels on both rows and columns
$<col><row>	$C4	Absolute reference on columns, relative reference on rows.
<col>$<row>	C$4	Relative reference on columns, absolute reference on rows.
<col><row>	C4	Relative reference on columns relative reference on rows.
$<col>	$C	
$<row>	$4	

NOTATION FOR USE WITH MULTIPLE HIERARCHIES

When the $ notation is used to indicate an absolute reference to a row or column it will only apply to the innermost hierarchy of the axis. If you want to specify an absolute reference to more than one hierarchy then you need to include a $ for each hierarchy in the query. The following example explains this:

	A	B	C	D	E	F	G	H	I
1		Executive General and Administration				Inventory Management			
2		North America Operations		AdventureWorks Cycle		North America Operations		Adventure eWorks Cycle	
3		Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget
4	Operating Profit	-397,061	-145,800	-397,061	-145,800	-1,316,774	-412,800	-1,316,774	-412,800
5	Other Income and Expense	746		746		2,449		2,449	
6	Taxes	387		387		1,069		1,069	
7	Net Income	-396,702	-145,800	-396,702	-145,800	-1,315,394	-412,800	-1,315,394	-412,800

Reference	Meaning
$B	Refers to the column “Actual” wherever it appears in the grid
$$B	Refers to the combination of “Actual” and “North America” wherever it appears in the grid
$$$B	Refers to the combination of “Actual”, “North America” and “Executive General and Administration” which only appears in column B

ADDITIONAL NOTATIONS

MDXL adds one additional construct which is the use of the # symbol to allow referencing of the current member in a hierarchy based on the hierarchies displayed in the result set.

Notation	Example	Description
#<row/col>	#A	Refers to the “current member” in a hierarchy. This mode of referencing is only available for the first rows and columns in the grid which relate to the label cells.

iT-Workplace Limited
MWB Business Exchange
15 Wheeler Gate
Nottingham NG1 2NA
United Kingdom

Tel: +44 (0)115 852 4711
Web: http://www.it-workplace.com