

DeltaMaster clicks!

05/2009

Greetings, fellow data analysts!

Sometimes people use a derogatory tone when they talk or write about textbook knowledge. We, on the other hand, feel that a good textbook can offer very useful advice. In the case of visualization, you can even start with a grade school textbook. Nowadays, 7th graders are already learning how they should view area charts*.

By the 5th grade, many children have already learned about set theory, which shows how useful it can be to collect individual objects into groups. That way, you can simply refer to the set instead of listing every single member. Economists and EU Economic Ministries know the benefits of sets all too well. After all, if they hadn't defined M3 as the money supply for the Euro zone, how much time would they have to talk about it if they had to constantly say "currency in circulation + overnight deposits + deposits with an agreed maturity of ... + deposits redeemable at a notice period of ... + repurchase agreements, shares of money market funds and debt securities up to..."?

These types of constructs are useful in reporting as well. MDX, for example, supports named sets. And using them with *DeltaMaster* is child's play. Taking this edition of *clicks!* as your guide, why not try it out for yourself? We're sure that you will treasure named sets and the time you'll gain – for having insightful discussions or simply enjoying a good book.

* <http://www.bella-consults.com/competent-media>

Best regards,
Your Bissantz & Company team

Executive Forum 'From Guerilla Controlling to Management Intelligence'

25 May 2009, Berlin

Final call! There are only a few seats left for our seminar discussing current and upcoming developments in management information. Join us as pioneers in this field shed insight on their strategies and next steps!
www.bissantz.com/executive

5th Handelsblatt CFO Congress 28-29 May 2009, Munich

We're there! And you?
www.cfo-kongress.de

DeltaMaster@Work

25 June 2009, Nuremberg

All seats are already taken for the May seminar! So register today for the June event or request a special session.
www.bissantz.com/dm@w

Bissantz Campus

Our training offerings for *DeltaMaster* and Microsoft SQL Server/Analysis Services
www.bissantz-campus.de

Archive

www.bissantz.com/clicks/en

DeltaMaster Matinee on 2 April 2009 in Cologne

Thanks again to Daniel Taudien (right) who captivated our audience with his presentation on how the ColorExpert-Storch Group uses *DeltaMaster*. By the way...our next matinee will take place on 16 July 2009 in Stuttgart.
www.bissantz.com/matinee



Tip of the month Reusing axis definitions with named sets

For quite some time now, we have been promoting the concept of ‘industrial reporting’ in order to make creating reports even more efficient. Our inspiration stemmed from disciplines such as engineering and computer science, where ‘recycling’ is a key construction/development goal for quality assurance and more cost-efficient production/programming.

You can already reuse your work in *DeltaMaster* in many different areas. In this edition of *clicks!*, however, we will solely focus on named sets. This useful function saves you time, for example, when you want to create new pivot tables, calculated members or user-defined measures. In addition, *ReportServer* helps you generate individual reports for a clearly defined set of measures.

Named sets – an abstract description

A named set is an MDX query expression that returns a group of dimension members or measure values. Named sets generally use practical – and not technical – names.

Common examples include customers or articles that are dynamically selected based on certain criteria.

Seat Center	23.323	53.023	-29.700	-56,0%
Masters	19.608	23.926	-4.318	-18,0%
System & Plan	15.940	48.341	-32.400	-67,0%
Magna Lumina	14.865	16.563	-1.698	-10,3%
Sum	688.387	1.313.033	-624.647	

Customers ordered descendingly by Revenues, Revenues, Current >= 10.000 and Revenues, Dev. PrevPer < 0,00; 14 rows,
DeltaMaster powered by Bissantz™

Using that name, you can easily reuse the expression in different queries, for example, in multiple pivot tables or other MDX calculations. If you later modify the definition, these changes will then automatically apply everywhere the name is used. The term ‘named set’ stems from the MDX programming language. Some databases vendors, however, have created their own names for this functionality. While Oracle calls them ‘value sets’, Infor PM OLAP (MIS Alea) and IBM Cognos (Applix) TM1 simply refer to them as ‘subsets’.

In *DeltaMaster*, you can create named sets in *Pivotizer* level or higher; this will be the main focus of this edition of *clicks!*. *DeltaMaster*, however, also automatically reads existing named sets from many different types of OLAP databases in your analysis model or session. As a result, you can use a subset e.g. defined in a TM1 database for your *DeltaMaster* analyses and reports without any manual work.

If that sounds rather abstract, that’s good because it was intentional. You need to apply a bit of abstract thinking to recycle named sets. Now we’ll get more concrete and show a few examples of how you can work with named sets.

Creating a named set

Let's assume that you want to keep an eye on a certain group of customers – namely, those that had noteworthy revenues (e.g. larger than 10,000) in the previous period but haven't yet achieved that volume in the current period. In other words, you are looking for customers with a negative revenue variance in comparison to the previous period. In addition, you want to sort them in descending order based on their past revenues.

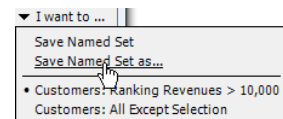
Revenues	...	Presentation	...	PrevPer	Dev. PrevPer	Dev. PrevPer %
Customers	...	03/2009				
Allberg Sys		169.632		209.263	-39.631	-18,9%
Robertson Dispatchers		90.927		167.679	-76.752	-45,8%
GoodChairs		75.715		164.349	-88.634	-53,9%
Summers Furniture		63.897		85.965	-22.068	-25,7%
Better Living		53.760				
Officina AG		46.864				
Table & Chair		31.789				
ArcNova		30.541				
Bavaria Furniture		26.156				
MFW		25.368				
Seat Center		23.323				
Masters		19.608				
System & Plan		15.940				
Magna Lumina		14.865				
Sum		688.387				

To create this type of pivot table, we need two filter values –the revenues and the revenue variance for the previous period. As a result, you edit the *Axis definition* of the customer dimension, define a *Filter*, and a *Ranking* on the respective tabs. On the *General* tab, we have activated the *Level selection* for the lowest level available: the individual customers.

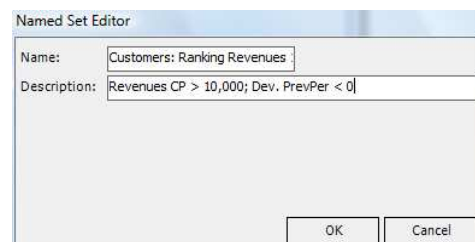
Now we admit that this is not a very complex query. But even in this simple example, imagine now that you wanted to run a budget-actual variance or evaluate the purchased product groups for these exact customers – or generate sales statistics for each individual customer. Would you really want to have to select all these criteria again? Not really! In these cases, it would be easier if you could just group this specific market segment and give it a name. And that is exactly what a named set does. It allows you to save the complete axis definition and dynamically reuse it in other pivot tables. 'Dynamically' in this case means that if you have saved a level selection with its own filter rules, this will remain a dynamic level selection with the same filter rules in another scenario as well and will not explicitly list the current query results.

Naming sets

You can define the options for named sets in the *Axis definition*, which is located in the upper right part of your screen under the *I want to...* menu. Here, you can save the new set so that you can reuse it again and again.



In the *Named Set editor* (simplified here in the screenshot on your right), you simply enter its *Name* and *Description*. You can use spaces and special characters in both fields because they are primarily used for displaying descriptions in dialog boxes, menus or in the MDX editor, a special *DeltaMaster* feature which is available everywhere where you can use MDX expressions. Since the MDX editor automatically replaces your given names internally with technical expressions, you can use normal English language and syntax in these fields.



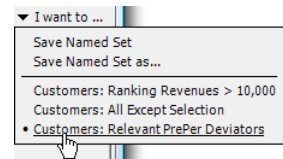
As soon as you have saved the definition as a named set, this name will appear below the *I want to...* menu. That way, you can instantly see if the properties in the *General*, *Ranking*, *Filter*, *Member property* and *Options* tabs only apply for the current axis definition or if they stem from a named set.



In addition to a *Level selection* as shown in this example, you can also save the definitions of a *Member selection* or a *user-defined MDX expression*.

Reusing named sets in a pivot table

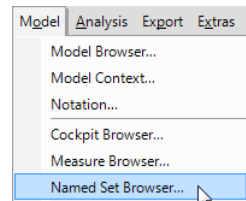
You can easily access your named sets from the *I want to...* menu. *DeltaMaster* will then automatically apply the parameters that were defined in the named set to the current axis definition. This way, you can quickly and easily reuse an available selection in a new pivot table. Of course, if you used a named set to configure the *Axis definition*, you can still modify these settings for that particular pivot table. If you change the settings, however, the axis definition will be disconnected from the set and will not reflect any later changes made to the set definition.



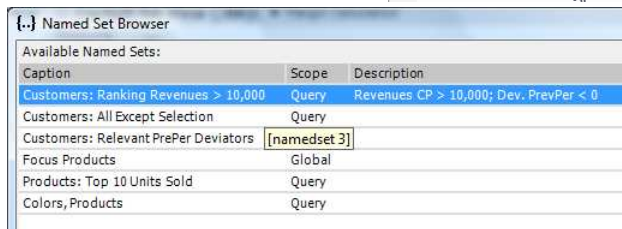
Since each axis of a pivot table renders a single dimension, the *Axis definition* only supports named sets that use a single, related dimension. As a result, *DeltaMaster* will only display sets in the *I want to...* menu if they match the dimension shown in the current axis. In other words, if you have also saved sets from the product dimension, *DeltaMaster* will not display them if you are working on an *Axis definition* for the customer dimension. (Generally speaking, however, you can create named sets that incorporate over multiple dimensions.)

Administration with the named set browser

The *Named set browser* (see *Model* menu) provides an overview of all named sets that are available in both *DeltaMaster* and the OLAP database. Here, you can *Rename* and *Delete* named sets as well as *Change* their *description* or *Properties* (context menu, *I want to...* menu).



If you hold the *Alt* key and mouse over a set, a tooltip will display the automatically generated technical name (see '[namedset 3]' in the screenshot on your right) which you can use in your own MDX expressions.

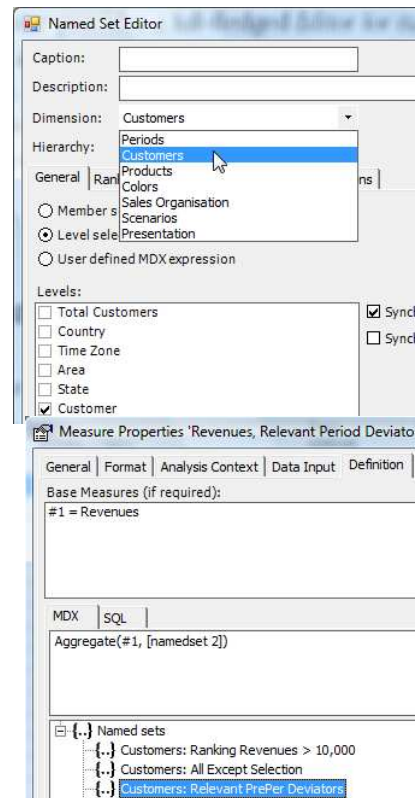


The 'Focus Products' shown in the screenshot above originates from the database, which is designated in the *Scope* field with the term 'Global'. Although you cannot modify this set in *DeltaMaster*, you can view its exact definition under *Properties* (context menu, *I want to...* menu). Under *Properties*, you can also view a set's *Dependencies*, for example, which other calculated objects it uses ('parts list'), which other calculated objects they use ('usage history') as well as where (i.e. in which cockpits, analysis templates, reports, and views) it appears.

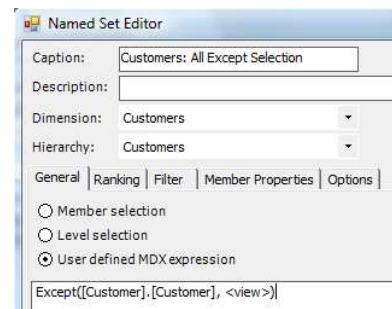
If you want to *Edit* or *Add* named sets in the *Named set browser*, *DeltaMaster* provides a full-fledged *Named Set Editor*. Its look, feel and functionality are very similar to that of the *Axis definition*. The main difference, however, is you can select the dimension in which the set definition should apply in the *Named Set Editor*.

Reusing named sets in MDX expressions

You can also reuse your MDX expressions to create user-defined measures, calculated members and user-defined MDX expressions. Using the customer selection that we defined on page 3, let's assume that we want to create a new measure that lists these customers' total revenues regardless of what view is displayed. You can do that with a user-defined measure as shown in the screenshot on your right. The MDX expression 'Aggregate([namedset 2], #1)' uses the *Aggregate()* function to create that sum. As a function parameter, use the 'Revenues' measure which you want to add together and the members with relevant revenues. And that is the definition of the named set which we previously defined in the pivot table. Best of all, you don't need to enter the technical name '[namedset 2]'. you simply drag the 'Customers: Relevante PrevPer Deviators' entry from the lower part of the MDX editor and drop it into the *Description* field. *DeltaMaster* will then automatically add the relevant identifier.

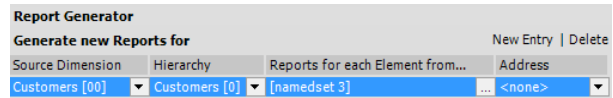


This next example shows that named sets don't just occur in other MDX expressions – you can also use them to create user-defined MDX expressions! The set called 'Customers: All Except Selection' is defined as 'Except([Customer].[Customer], <view>}'. This statement will search the customer dimension and return all members except for those which are already selected in the *View* window. Why not add this set to your application right now? An inverse view of your selection often comes in handy.

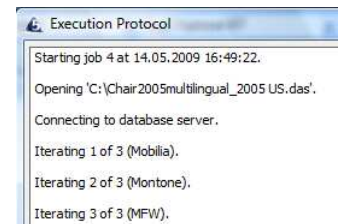


Reusing named sets in ReportServer

You can also use named sets in the *Report Generator* function of *ReportServer*. Its job is to 'iterate', or sift, through all of the members for which it should create a separate report. Since the *Generate New Reports for* field (which you use to define iterations) also supports MDX statements, you can use named sets here as well. Please check, however, that the selected *Source dimension* matches the named set.



In this example, you can easily create current, detailed reports on every single customer that is particularly important for you right now.



Questions? Comments?

Just contact your Bissantz team for more information!